日　本　国　特　許　庁
JAPAN　PATENT　OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application:　２００１年　４月１８日

出願番号
Application Number:　特願２００１－１２０３３５

[ST.10/C]:　[ＪＰ２００１－１２０３３５]

出願人
Applicant(s):　大森　聡

２００３年　３月１４日

特許庁長官
Commissioner,
Japan Patent Office　太田信一郎

出証番号　出証特２００３－３０１７０２２

【書類名】　特許願

【整理番号】　2001A09

【提出日】　平成13年 4月18日

【あて先】　特許庁長官殿

【発明者】

【住所又は居所】　埼玉県浦和市西堀４丁目１１番７号６２７

【氏名】　大森　聡

【特許出願人】

【識別番号】　300000513

【氏名又は名称】　大森　聡

【先の出願に基づく優先権主張】

【出願番号】　特願2000-117343

【出願日】　平成12年 4月19日

【先の出願に基づく優先権主張】

【出願番号】　特願2000-149122

【出願日】　平成12年 5月19日

【手数料の表示】

【予納台帳番号】　095958

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　ヌクレオチド等の配列情報の記録方法及び装置、前記配列情報の供給方法、前記配列情報を記録した記録媒体、並びに要約値の計算方法

【特許請求の範囲】

【請求項１】　一列のヌクレオチドの配列情報の記録方法であって、

前記一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、前記一列のヌクレオチドの配列に関する情報を記録することを特徴とするヌクレオチドの配列情報の記録方法。

【請求項２】　請求項１記載の記録方法であって、

前記一列のヌクレオチドは４種類のヌクレオチドよりなり、

前記４種類のヌクレオチドを互いに異なる６ビット以下のデータで表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項３】　請求項２記載の記録方法であって、

前記４種類のヌクレオチドを互いに異なる２ビットのデータで表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項４】　請求項２、又は３記載の記録方法であって、

前記一列のヌクレオチドは、一つのＤＮＡを構成する１対の重合体の鎖の内の１本の鎖の全部又は一部であり、

前記４種類のヌクレオチド中の互いに相補的な２対のヌクレオチドをそれぞれ互いにビット反転の関係にある１対のデータで表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項５】　請求項２、又は３記載の記録方法であって、

前記一列のヌクレオチドは、一つのＲＮＡを構成する一つの重合体の鎖の全部又は一部であることを特徴とするヌクレオチドの配列情報の記録方法。

【請求項６】　請求項１記載の記録方法であって、

前記一列のヌクレオチドの配列に関する情報を、前記配列を表すテキストデータ又は数値データの数学的な要約値で表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項７】　請求項６記載の記録方法であって、

前記一列のヌクレオチドは２５個以上のヌクレオチドの配列であり、

前記一列のヌクレオチドの配列に関する情報を４０ビット以上で１９２ビット以下の長さの数学的な要約値で表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項８】　請求項７記載の記録方法であって、

前記数学的な要約値は、前記一列のヌクレオチドの配列に対応するテキストデータ又は数値データにＭＤ５ハッシュ関数、又はＳＨＳハッシュ関数の演算を施して得られることを特徴とするヌクレオチドの配列情報の記録方法。

【請求項９】　請求項１～６の何れか一項記載の記録方法であって、

前記一列のヌクレオチドは遺伝子の少なくとも一部であることを特徴とするヌクレオチドの配列情報の記録方法。

【請求項１０】　請求項１記載の記録方法であって、

前記一列のヌクレオチドの配列に対応するテキストデータを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる６ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第１の演算を施して第１組のシンドローム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第２の演算を施して第２組のシンドローム情報を求め、

前記第１組及び第２組のシンドローム情報で前記一列のヌクレオチドの配列を表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項１１】　請求項１０記載の記録方法であって、

複数行の前記変換データの各行の変換データをそれぞれ前記非配列方向に交互に第１群の変換データ及び第２群の変換データに分けたとき、

前記第１の演算は、所定の整数Ｋを用いて前記第１群の変換データ、及び前記第２群の変換データのそれぞれの法Ｋのもとの和を求める演算であり、

前記第２の演算は、複数列の前記変換データの各列の変換データに対する法Ｋのもとの和を求める演算であることを特徴とするヌクレオチドの配列情報の記録方法。

【請求項１２】　請求項１０、又は１１記載の記録方法であって、

前記一列のヌクレオチドの配列を基準配列として、該基準配列の２組の前記シンドローム情報に対応させて、検査対象の一列のヌクレオチドの配列の２組のシンドローム情報を求め、

前記４組のシンドローム情報より前記基準配列に対する前記検査対象の一列のヌクレオチドの配列の相違部を求めることを特徴とするヌクレオチドの配列情報の記録方法。

【請求項１３】　一列のヌクレオチドの配列情報の記録装置であって、

一つの核酸の少なくとも一部に含まれる一列のヌクレオチドの配列情報を読み取る配列読み取り装置と、

該配列読み取り装置で読み取られた配列の情報をテキストデータとして第１ファイルに記録する第１記録手段と、

前記第１ファイルのテキストデータよりも少ないデータ量で、前記配列読み取り装置で読み取られた配列の情報を表し、該配列の情報を第２ファイルに記録する第２記録手段と

を有することを特徴とするヌクレオチドの配列情報の記録装置。

【請求項１４】　請求項１３記載の記録装置であって、

前記第２記録手段は、前記配列読み取り装置で読み取られた一列のヌクレオチドの配列を、該配列を表すテキストデータ又は数値データの数学的な要約値で表すことを特徴とするヌクレオチドの配列情報の記録装置。

【請求項１５】　請求項１３記載の記録装置であって、

前記第２記録手段は、前記配列読み取り装置で読み取られた一列のヌクレオチドの配列に対応するテキストデータを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに

異なる６ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第１の演算を施して第１組のシンドローム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第２の演算を施して第２組のシンドローム情報を求め、

前記第１組及び第２組のシンドローム情報を前記第２ファイルに記録することを特徴とするヌクレオチドの配列情報の記録装置。

【請求項１６】　一列のヌクレオチドの配列情報を記録したコンピュータ読み取り可能な記録媒体であって、

前記一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、前記一列のヌクレオチドの配列に関する情報が記録されたことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項１７】　請求項１６記載の記録媒体であって、

前記一列のヌクレオチドは２５個以上のヌクレオチドの配列であり、

前記一列のヌクレオチドの配列に関する情報は、４０ビット以上で１９２ビット以下の長さの数学的な要約値で前記記録媒体に記録されたことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項１８】　請求項１６記載の記録媒体であって、

前記一列のヌクレオチドの配列に対応するテキストデータを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる６ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第１の演算を施して第１組のシンドローム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第２の演算を施して第２組のシンドローム情報を求めておき、

前記一列のヌクレオチドの配列に関する情報は、前記第１組及び第２組のシンドローム情報として前記記録媒体に記録されたことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項１９】　一列のヌクレオチドの配列情報の供給方法であって、

前記一列のヌクレオチドの配列に対応するテキストデータ、又は複数種類のヌクレオチドに対して互いに異なる６ビット以下の数値データを割り当てることによって前記テキストデータを変換して得られる数値データを保持する供給者が、

前記一列のヌクレオチドの配列の長さの情報、及び前記配列を表すテキストデータ又は前記数値データの数学的な要約値の情報を通信回線を介して閲覧可能な状態にしておき、

前記通信回線を介して前記配列の長さの情報及び前記数学的な要約値の情報を閲覧したユーザより、前記テキストデータ又は前記数値データの少なくとも一部の情報に対する取得要求が前記供給者に届いた後に、

前記供給者が前記ユーザに前記テキストデータ又は前記数値データの少なくとも一部の情報を供給することを特徴とするヌクレオチドの配列情報の供給方法。

【請求項２０】　請求項１９記載の供給方法であって、

前記一列のヌクレオチドは２５個以上のヌクレオチドの配列であり、

前記数学的な要約値は、４０ビット以上で１９２ビット以下のデータであり、

前記供給者は、更に前記一列のヌクレオチドの所定の一部の配列の情報を通信回線を介して閲覧可能な状態にしておくことを特徴とするヌクレオチドの配列情報の供給方法。

【請求項２１】　請求項１９、又は２０記載の供給方法であって、

前記供給者は、前記一列のヌクレオチドの配列に対応するテキストデータ、又はこれに対応する前記数値データを第１ファイルに記録して保持し、

前記供給者は、前記テキストデータ、又は前記数値データを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分データに分割し、

前記部分データを、それぞれ複数種類のヌクレオチドに対して互いに異なる６ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第１の演算を施して第１組のシンドローム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第２の演算を施して第２組のシンドローム情報を求め、

前記第１組及び第２組のシンドローム情報を第２ファイルに記録して保持し、

第１段階として前記ユーザは、前記供給者より前記第２ファイルに記録されている２組のシンドローム情報を受け取り、

前記２組のシンドローム情報に基づいて検査対象の一列のヌクレオチドの配列の内の前記供給者の一列のヌクレオチドの配列との相違部を特定し、

該相違部の配列の復元ができない場合に、第２段階として前記ユーザは前記供給者より前記第１ファイルに記録されている前記テキストデータ、又は前記数値データの内の前記配列の復元ができない部分の情報の提供を要求することを特徴とするヌクレオチドの配列情報の供給方法。

【請求項２２】　一列のアミノ酸の配列情報の記録方法であって、

前記一列のアミノ酸の配列に対応するテキストデータよりも少ないデータ量で、前記一列のアミノ酸の配列に関する情報を記録することを特徴とするアミノ酸の配列情報の記録方法。

【請求項２３】　請求項２２記載の記録方法であって、

前記一列のアミノ酸は、一つのタンパク質を構成する１本のアミノ酸の鎖の全部又は一部であり、

前記一列のアミノ酸の配列に対応するテキストデータを、２０種類のアミノ酸に対して互いに異なる６ビット以下のデータを割り当てることによって変換することを特徴とするアミノ酸の配列情報の記録方法。

【請求項２４】　請求項２２記載の記録方法であって、

前記一列のアミノ酸の配列に関する情報を、前記配列を表すテキストデータの数学的な要約値で表すことを特徴とするアミノ酸の配列情報の記録方法。

【請求項２５】　請求項２４記載の記録方法であって、

前記一列のアミノ酸は２５個以上のアミノ酸の配列であり、

前記一列のアミノ酸の配列に関する情報を１６ビット以上で１９２ビット以下

の長さの数学的な要約値で表すことを特徴とするアミノ酸の配列情報の記録方法。

【請求項２６】　請求項２４、又は２５記載の記録方法であって、

前記数学的な要約値は、前記一列のアミノ酸の配列に対応するテキストデータにＭＤ５ハッシュ関数、又はＳＨＳハッシュ関数の演算を施して得られることを特徴とするアミノ酸の配列情報の記録方法。

【請求項２７】　請求項２２記載の記録方法であって、

前記一列のアミノ酸の配列に対応するテキストデータを、前記アミノ酸の配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のアミノ酸に対して互いに異なる８ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第１の演算を施して第１組のシンドローム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第２の演算を施して第２組のシンドローム情報を求め、

前記第１組及び第２組のシンドローム情報で前記一列のアミノ酸の配列を表すことを特徴とするアミノ酸の配列情報の記録方法。

【請求項２８】　請求項２７記載の記録方法であって、

複数行の前記変換データの各行の変換データをそれぞれ前記非配列方向に交互に第１群の変換データ及び第２群の変換データに分けたとき、

前記第１の演算は、所定の整数Ｋを用いて前記第１群の変換データ、及び前記第２群の変換データのそれぞれの法Ｋのもとの和を求める演算であり、

前記第２の演算は、複数列の前記変換データの各列の変換データに対する法Ｋのもとの和を求める演算であることを特徴とするアミノ酸の配列情報の記録方法。

【請求項２９】　一列のアミノ酸の配列情報の記録装置であって、

一つのタンパク質の少なくとも一部に含まれる一列のアミノ酸の配列情報をテキストデータとして第１ファイルに記録する第１記録手段と、

前記第１ファイルのテキストデータよりも少ないデータ量で、前記一列のアミノ酸の配列の情報を表し、該配列の情報を第２ファイルに記録する第２記録手段と

を有することを特徴とするアミノ酸の配列情報の記録装置。

【請求項３０】　請求項２９記載の記録装置であって、

前記第２記録手段は、前記一列のアミノ酸の配列を、該配列を表すテキストデータの数学的な要約値で表すことを特徴とするアミノ酸の配列情報の記録装置。

【請求項３１】　一列のアミノ酸の配列情報の供給方法であって、

前記一列のアミノ酸の配列に対応するテキストデータ、又は複数種類のアミノ酸に対して互いに異なる８ビット以下の数値データを割り当てることによって前記テキストデータを変換して得られる数値データを保持する供給者が、

前記一列のアミノ酸の配列の長さの情報、及び前記配列を表すテキストデータ又は前記数値データの数学的な要約値の情報を通信回線を介して閲覧可能な状態にしておき、

前記通信回線を介して前記配列の長さの情報及び前記数学的な要約値の情報を閲覧したユーザより、前記テキストデータ又は前記数値データの少なくとも一部の情報に対する取得要求が前記供給者に届いた後に、

前記供給者が前記ユーザに前記テキストデータ又は前記数値データの少なくとも一部の情報を供給することを特徴とするアミノ酸の配列情報の供給方法。

【請求項３２】　請求項３１記載の供給方法であって、

前記一列のアミノ酸は２５個以上のアミノ酸の配列であり、

前記数学的な要約値は、１６ビット以上で１９２ビット以下のデータであることを特徴とするアミノ酸の配列情報の供給方法。

【請求項３３】　請求項３１、又は３２記載の供給方法であって、

前記供給者は、前記一列のアミノ酸の配列に対応するテキストデータ、又はこれに対応する前記数値データを第１ファイルに記録して保持し、

前記供給者は、前記テキストデータ、又は前記数値データを、前記アミノ酸の配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分データに分割し、

前記部分データを、それぞれ複数種類のアミノ酸に対して互いに異なる８ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第１の演算を施して第１組のシンドローム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第２の演算を施して第２組のシンドローム情報を求め、

前記第１組及び第２組のシンドローム情報を第２ファイルに記録して保持し、

第１段階として前記ユーザは、前記供給者より前記第２ファイルに記録されている２組のシンドローム情報を受け取り、

前記２組のシンドローム情報に基づいて検査対象の一列のアミノ酸の配列の内の前記供給者の一列のアミノ酸の配列との相違部を特定し、

該相違部の配列の復元ができない場合に、第２段階として前記ユーザは前記供給者より前記第１ファイルに記録されている前記テキストデータ、又は前記数値データの情報の提供を前記供給者に要求することを特徴とするアミノ酸の配列情報の供給方法。

【請求項３４】　一つ又は複数のファイルに記録されたデータの要約値を計算するための要約値の計算方法であって、

前記一つ又は複数のファイルに記録されたデータの内で所定のコードを無視して要約値を計算することを特徴とする要約値の計算方法。

【請求項３５】　請求項３４記載の要約値の計算方法であって、

前記無視する所定のコードは、数字コード、スペースコード、及び改行コードであることを特徴とする要約値の計算方法。

【請求項３６】　請求項３４記載の要約値の計算方法であって、

前記無視する所定のコードは、同一又は互いに異なる２組のコード、及びこれら２組のコードに挟まれたデータであることを特徴とする要約値の計算方法。

【請求項３７】　請求項３４記載の要約値の計算方法であって、

前記一つ又は複数のファイルから１文字分のコードデータを読み出す毎に、

該読み出されたコードデータが前記所定のコードであるときには、該読み出されたコードデータを無視して、次の１文字分のコードデータの読み出しを行い、

該読み出しによって得られた前記所定のコード以外のコードデータが予め定められた個数になるか、又は読み出すべきデータがなくなったときに、要約値の計算を行うことを特徴とする要約値の計算方法。

【請求項３８】　一連のテキストデータの要約値を計算するための要約値の計算方法であって、

前記一連のテキストデータを先頭から順に所定個数ずつのコードデータを含む複数の部分テキストデータと、前記所定個数よりも少ない個数のコードデータを含む端数のテキストデータとに分割し、

前記複数の部分テキストデータ、及び端数のテキストデータをそれぞれ分割する順序を含むデータとともに互いに異なる複数のファイルに記録し、

該複数のファイルに記録されたテキストデータから分割の順序に従って順次要約値を計算することを特徴とする要約値の計算方法。

【請求項３９】　請求項３８記載の要約値の計算方法であって、

前記所定個数ずつのコードデータ、及び前記所定個数よりも少ない個数のコードデータからは、所定のコードデータが除外されていることを特徴とする要約値の計算方法。

【請求項４０】　請求項３９記載の要約値の計算方法であって、

前記所定個数は、要約値を計算する際のデータ量の単位に応じて定められることを特徴とする要約値の計算方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、例えばＤＮＡ（デオキシリボ核酸：deoxyribonucleic acid)又はＲＮＡ（リボ核酸：ribonucleic acid）等の核酸の少なくとも一部を構成する一列のヌクレオチドの配列情報、又はタンパク質の少なくとも一部を構成する一列のアミノ酸の配列情報の記録方法及び装置に関する。更に本発明は、その配列情報を供給するためのビジネスモデルとして好適な配列情報の供給方法、その配列情報を記録したコンピュータ読み取り可能な記録媒体、及びその記録方法を実施する際に使用できる要約値の計算方法に関する。

【０００２】

【従来の技術】

人間、及び他の生物（動物、植物、微生物等）のＤＮＡを構成する１対のヌクレオチドの鎖（又は塩基の鎖）の配列情報の解読が世界的に行われている。この場合、従来よりＤＮＡを構成する４種類のヌクレオチドは、塩基としてアデニンを含むヌクレオチド、グアニンを含むヌクレオチド、シトシンを含むヌクレオチド、及びチミンを含むヌクレオチドにそれぞれ文字Ａ，Ｇ，Ｃ，及びＴを割り当てることによって、それぞれ１バイト（＝８ビット）のテキストデータで表わされている。その結果として一つのＤＮＡの配列は、それを構成する１対の重合体の鎖の内の一方の鎖のヌクレオチド（ｎ個とする）の配列を順次文字Ａ，Ｇ，Ｃ，Ｔ（又はａ，ｇ，ｃ，ｔ）の何れかで表すことによって、ｎバイトのテキストデータで表されていた。同様に、一つのＲＮＡを構成する１本のｎ個のヌクレオチドの配列は、チミンを含むヌクレオチドの代わりにウラシルを含むヌクレオチドに文字Ｕ（又はｕ）を割り当てることによって、ｎバイトのテキストデータで表されていた。

【０００３】

これに関して、例えば人間の最も大きい第１染色体中のＤＮＡの配列は、約２億５千万個のヌクレオチドの配列であり、最も小さい第２２染色体中のＤＮＡの配列は、約５０００万個のヌクレオチドの配列であるため、人間の各染色体中のＤＮＡの配列は、約２５０Ｍバイト～５０Ｍバイトのテキストデータで表すことができる。更に、一人の人間の全部のＤＮＡ情報（ゲノム）は、約３０億個のヌクレオチドの配列で表すことができるため、そのゲノムは、約３Ｇバイトのテキストデータで記録することができる。なお、それらのテキストデータに対して通常のファイル圧縮技術を適用することによって、それらのテキストデータは、例えば元のデータの５０％程度の圧縮ファイルとしても記録、又は送信することができる。

【０００４】

また、ＤＮＡの配列の解読に続いて、ＤＮＡ中の多数の遺伝子の情報に基づいてそれぞれ合成されるタンパク質の機能の研究も広く行われている。この場合、

タンパク質を構成する２０種類のアミノ酸は、三文字表記（3-Letter Code）ではそれぞれ３文字（例えばＡｌａ，Ｃｙｓ，Ｇｌｕ等）のテキストデータで表され、一文字表記（1-Letter Code）ではそれぞれ１文字のテキストデータ（例えばＡ，Ｃ，Ｅ等）で表されるため、ｎ個のアミノ酸よりなるタンパク質の配列は、ｎバイトのテキストデータで表すことができる。そして、種々のタンパク質は、それらのアミノ酸が約２０個～約１０００個程度所定の順序で配列されたものであるため、それらのタンパク質の配列は、最大でも約１ｋバイト程度のテキストデータで記録することができる。また、例えば人間の遺伝子の総数は約３万個と言われており、タンパク質は理論的なものも含めて約１０万種類の存在が可能であると言われている。

【０００５】

【発明が解決しようとする課題】

上記の如く例えば一人の人間のＤＮＡ情報をテキストデータで記録するためには、全部で３Ｇバイト程度の記憶容量が必要であり、仮に通常の圧縮ファイルの技術を適用しても１Ｇバイト程度の記憶容量が必要である。また、人間以外の大腸菌や各種ウィルス等のＤＮＡ情報も解析されて次第に公開されるようになっているが、これらのＤＮＡ情報をテキストデータの形で多く集めると、数１００Ｍバイド程度の記憶容量が必要である。これはＲＮＡの配列情報についても同様である。

【０００６】

このように人間又は他の生物のＤＮＡ情報をテキストデータ、又はこの通常の圧縮ファイルの形で記録するものとすると、例えば１枚の記憶容量が５Ｇバイト程度のＤＶＤ－ＲＯＭ(digital video disc-ROM)ディスクのように膨大な記憶容量を持つ記録媒体が必要である。更に、そのＤＮＡ情報を利用する場合にその記録媒体からの読み出し時間が長くなり、処理時間が長くなるという不都合がある。

【０００７】

また、現状の一般の通信回線の通信速度は、最大で１Ｍｂｐｓ程度であるため、例えば１Ｇバイト程度のＤＮＡ情報をその通信回線を介して送信するものとす

ると、送信時間は最短でも約２時間程度となり、あまり実用的ではない。特に最近はそのＤＮＡ情報をデジタルの携帯電話システムを介して送信する場合も考えられるが、現在の携帯電話システムの通信速度はせいぜい１００ｋｂｐｓ程度であるため、少なくとも人間のＤＮＡ情報の伝送で使用することは困難である。

【０００８】

次に、例えば或る微生物のＤＮＡ中の遺伝子について複数の研究者が並行して研究するような場合に、複数の研究者が保有している標準となるＤＮＡのヌクレオチドの配列の同一性をどのように保証するのかという問題がある。即ち、そのＤＮＡのヌクレオチドの配列が例えば数Ｍバイト（文字数で数１００万文字）程度のテキストデータで記録されている場合に、複数の研究者が互いに自分のテキストデータと他人のテキストデータとの同一性（完全一致性）を短時間に確認するのは必ずしも容易ではない。

【０００９】

これに関連して、例えば人間又は他の生物のＤＮＡ情報の利用方法としては、標準的なＤＮＡの配列と、検査対象のＤＮＡの配列との間の相違する部分をサーチする場合が考えられる。これは、いわゆるＳＮＰ（一塩基変位多型：Single Nucleotide Polymorphism）の可能性を検査するような場合に必要になると考えられる。しかしながら、両方のＤＮＡのヌクレオチドの配列がそれぞれ膨大なテキストデータで表わされている場合に、それら２つのテキストデータを比較して相違点を検出するにはかなりの長い時間が必要となり、検査時間が長くなるという不都合がある。

【００１０】

更に、人間又は他の生物のＤＮＡ情報を製薬会社の研究者等のユーザに提供するビジネスも行われつつあるが、この場合に、複数の情報供給者間で重複した情報の提供をできるだけ避けることが望ましい。このためには、複数の情報供給者間で、ＤＮＡのヌクレオチドの全体の配列情報を公開することなく、ヌクレオチドの配列の同一性を容易に確認できるようにすることが望ましい。更に、情報供給者が例えば通信回線を介してＤＮＡ情報をユーザに提供する場合には、できるだけ少ない情報量で、即ち短い送信時間で必要な情報をユーザに提供できるビジ

ネスモデルが必要である。また、ユーザ側では、提供されたＤＮＡ情報に伝送エラー等が無いかどうかを容易に確認できることが望ましい。上記の各課題はＲＮＡのヌクレオチドの配列情報についても同様に当てはまるものである。

【００１１】

更に、一つのタンパク質のアミノ酸の配列は、最大でも約１ｋバイト程度のテキストデータで記録することができるが、タンパク質の種類は理論的に約１０万個程度にもなるため、全部のタンパク質の配列情報をテキストデータで表すと、全部のＤＮＡの配列情報程度の膨大な量となる。従って、個々のタンパク質の配列は、できるだけ少ない情報量で記録できることが望ましい。また、２つのタンパク質の配列情報の同一性を容易に確認できるシステムも必要である。

【００１２】

本発明は斯かる点に鑑み、核酸中の一列のヌクレオチドの配列情報、又はタンパク質中の一列のアミノ酸の配列情報をできるだけ少ないデータ量で記録できる記録方法及び記録装置を提供することを第１の目的とする。

また、本発明は、２つのヌクレオチドの配列情報同士、又は２つのアミノ酸の配列情報同士の同一性を少ないデータ量で高精度に確認できる記録方法及び記録装置を提供することを第２の目的とする。

【００１３】

更に本発明は、２つのヌクレオチドの配列情報の間の相違する部分を少ないデータ量で容易に検出できると共に、必要に応じてその相違する部分の情報を復元できる記録方法及び記録装置を提供することを第３の目的とする。

また、本発明は、一列のヌクレオチドの配列情報、又は一列のアミノ酸の配列情報を少ないデータ量でユーザに提供できるビジネスモデルを提供することを第４の目的とする。

【００１４】

更に本発明は、そのビジネスモデルにおいて、ユーザが提供された配列情報と情報供給者が保持している配列情報との同一性、又は相違する部分を少ないデータ量で容易に確認できるようにすることをも目的とする。

また、本発明は、ヌクレオチドの配列情報が少ないデータ量で記録されたコン

ピュータ読み取り可能な記録媒体を提供することをも目的とする。

【００１５】

また、本発明は、ヌクレオチド又はアミノ酸の配列情報を記録する場合に使用できる効率的な要約値の計算方法を提供することを目的とする。

【００１６】

【課題を解決するための手段】

本発明によるヌクレオチドの配列情報の記録方法は、一列のヌクレオチドの配列情報の記録方法であって、その一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、その一列のヌクレオチドの配列に関する情報を記録するものである。

【００１７】

斯かる本発明によれば、その一列のヌクレオチドは、例えばＤＮＡ（deoxyribonucleic acid）を構成する１対の重合体の鎖の一方の鎖の少なくとも一部、又はＲＮＡ（ribonucleic acid）を構成する１列の重合体の鎖の少なくとも一部である。そして、その一列のヌクレオチドの配列は、各ヌクレオチドに含まれる塩基の配列ともみなすことができる。そのヌクレオチドの配列が、そのテキストデータ以外のより少ないデータ量のファイルとして記録される。従って、記録媒体として、ＤＶＤ－ＲＯＭのような大容量の媒体の他に、ＣＤ－ＲＯＭ、及びフラッシュＲＯＭのような小容量でも通常のコンピュータで手軽に再生できる媒体を使用できる。

【００１８】

更に、少ないデータ量の配列情報であれば、通信回線を介して短時間に送信できるため、実質的に安価に配列情報の供給を行うことが可能となる。

本発明において、その一列のヌクレオチドは４種類のヌクレオチドよりなり、その４種類のヌクレオチドを互いに異なる６ビット以下のデータで表すことが望ましい。テキストデータ形式では、各ヌクレオチドは、それぞれ８ビットのアスキーコード（ASCII CODE）、即ち文字Ａ，Ｇ，Ｃ，Ｔ（又はＵ）の何れかで表されるため、各ヌクレオチドを６ビット以下のデータで表すことによって、データ量を減らすことができる。

【００１９】

なお、テキストデータが記録されたファイルが通常の圧縮技術（ＺＩＰファイル、ＬＨＡファイル等）で圧縮できるように、本発明のデータが記録されたファイルも更に通常の圧縮技術で圧縮して記録できることは言うまでもない。但し、圧縮されたファイルを使用する場合には、解凍作業が必要になり、最終的には元のファイルを復元する必要があるため、元のファイル自体のデータ量を減らしておくことは極めて有効である。

【００２０】

また、その４種類のヌクレオチドを互いに異なる２ビットのデータで表すことが望ましい。２ビットのデータによって、最も少ないデータ量で４種類のヌクレオチド（又は塩基）を表すことができる。

また、その一列のヌクレオチドが、一つのＤＮＡを構成する１対の重合体の鎖の内の１本の鎖の全部又は一部であるときに、その４種類のヌクレオチド中の互いに相補的な２対のヌクレオチドをそれぞれ互いにビット反転の関係にある１対のデータで表すことが望ましい。互いに相補的な２対のヌクレオチドとは、互いに相補的な２対の塩基と実質的に同じ意味である。ここで、２進数で表現した数ｋをbin(k)として、例えば図２のＤＮＡ（５）に示すように、アデニンを含むヌクレオチド（７Ａ）をbin(00)で表したとき、それに対して相補的なチミンを含むヌクレオチド（７Ｔ）をbin(11)で表す。更に、グアニンを含むヌクレオチド（７Ｇ）をbin(01)で表したとき、それに対して相補的なシトシンを含むヌクレオチド（７Ｃ）をbin(10)で表す。この結果、ＤＮＡ（５）の一方のヌクレオチドの鎖（６Ａ）がbin(0001101111‥)（＝ＢＮＡとする）で表されて、それと相補的な他方のヌクレオチドの鎖（６Ｂ）に対応する２進数のデータＢＮＢは、コンピュータによって２進数ＢＮＡをビット毎に反転するだけで極めて高速に求めることができる。

【００２１】

次に、本発明において、より具体的な第１の記録方法は、その一列のヌクレオチドの配列に関する情報を、その配列を表すテキストデータ又は数値データの数学的な要約値（message digest）で表すものである。この数学的な要約値は、暗

号理論において、送信ファイルの作成者の本人確認を行うために、送信ファイルに所定のハッシュ関数を施すことによって得られる値と数学的には同等のものである。しかしながら、本発明においては、一列のヌクレオチドの配列を表すデータ（原データ）の要約値を、例えば最先の解読者の主張や、２つの膨大な原データの同一性の確認に使用する点が本質的に異なっている。即ち、或るＤＮＡのヌクレオチドの配列を最初に解読した者が、その配列を示す原データの要約値を例えばインターネット上で公開することによって、原データを公開することなく最先に解読したことを主張できる。また、例えば情報供給者からＤＮＡの配列情報を購入したユーザは、購入した配列情報の要約値を、例えばインターネット上で公開されているそのＤＮＡの要約値と比較することによって、購入した配列情報の同一性を高い確率で確認できる。更に、複数の研究者が同一のＤＮＡについて研究を行う場合に、各研究者が保持しているＤＮＡのヌクレオチドの膨大な配列情報の長さ、及び要約値を求め、これらを比較することによって、研究対象の同一性を容易に高い確率で確認することができる。

【００２２】

この場合、その一列のヌクレオチドが２５個以上のヌクレオチドの配列であるときに、その一列のヌクレオチドの配列に関する情報を４０ビット以上で１９２ビット以下の長さの数学的な要約値で表すことが望ましい。２５個以上のヌクレオチドの配列のテキストデータは、２００ビット（＝２５・８ビット）以上になるため、その要約値を１９２ビット以下とすることで、テキストデータよりも少ないデータ量となる。また、特に処理単位が６４ビットのコンピュータを使用する場合には、要約値の長さは６４ビットの倍数、即ち６４ビット、１２８ビット、又は１９２ビットが望ましいと考えられる。

【００２３】

また、例えば将来的に全人類のＤＮＡの配列情報を必要に応じて解読するような状況を想定して、世界人口を１００億人程度と仮定すると、そのＤＮＡの配列情報は約10^{10}通りにもなる。更に、安全係数を１００倍程度とすると、その要約値は、10^{12}（＝10^{10}・１００）通り、即ち約$2^{39.86}$通り以上の値を取る必要がある。このためには、その要約値を４０ビット以上の長さとすればよい。

これによって、２つのＤＮＡ又はＲＮＡの配列情報同士の同一性を10^{-12}以上の精度で高精度に確認できる。

【００２４】

更に、その数学的な要約値は、その一列のヌクレオチドの配列に対応するテキストデータ又は数値データにＭＤ５ハッシュ関数、又はＳＨＳ（Secure Hash Standard）ハッシュ関数の演算を施して得ることができる。この場合、ＭＤ５ハッシュ関数は、高速演算が可能であると共に、得られる要約値が１２８ビットであり、通常のコンピュータで処理し易い利点がある。一方、ＳＨＳハッシュ関数は、元のデータの推定がより困難であるが、得られる要約値が１６０ビットと、通常のＤＮＡ又はＲＮＡのヌクレオチドの配列の表現に関しては必要以上に長いと考えられる。従って、通常のヌクレオチドの配列の表現については、ＭＤ５ハッシュ関数がより実用的と考えられる。

【００２５】

また、暗号理論で使用されるハッシュ関数は、送信ファイルの内容が推定されないように、かつ内容の衝突の確率が極めて低くなるように設計されるため、その要約値は例えば最低でも１２８ビット程度の長さが必要とされると共に、複雑な演算が繰り返して実行される。これに対して本発明で使用するハッシュ関数は、通常の互いに異なるヌクレオチドの配列に対してほぼ衝突が無ければよいため、あまり複雑な演算を繰り返して行う必要は無いと考えられる。但し、通常の暗号理論で要約値の演算対象となるファイルは、せいぜい１Ｍバイト程度の長さであるのに対して、本発明で使用するハッシュ関数の演算対象は、例えば人間のＤＮＡのヌクレオチドの配列とすると、１００Ｍバイト程度にも達する膨大なデータのファイルである。そこで、本発明で使用するハッシュ関数（ハッシュ演算プログラム）は、演算対象の原ファイルを分割した後の複数の分割ファイルを順次処理することによって、全体の要約値を算出する機能を持つことが望ましい。

【００２６】

次に、本発明において、より具体的な第２の記録方法は、その一列のヌクレオチドの配列に対応するテキストデータを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータＴ（

ｉ，ｊ）に分割し、その部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる６ビット以下の数値データを割り当てることによって変換データＡ（ｉ，ｊ）に変換し、複数行のその変換データに各行毎にその非配列方向に第１の演算を施して第１組のシンドローム（syndrome）情報Ｂ１（ｉ），Ｂ２（ｉ）を求めると共に、複数列のその変換データに各列毎にその配列方向に第２の演算を施して第２組のシンドローム情報Ｃ（ｊ）を求め、その第１組及び第２組のシンドローム情報でその一列のヌクレオチドの配列を表すものである。

【００２７】

本発明においては、テキストデータを複数行で複数列の部分テキストデータに分割した後に、各部分テキストデータをそれぞれ変換データに変換しているが、これは予めそのテキストデータを一列の数値データに変換した後に、その数値データを複数行で複数列の変換データに分割することと実質的に等価である。本発明によれば、例えば図７に示す部分テキストデータＴ（ｉ，ｊ）を集めたテキストデータの情報の大部分を、例えば図９に示す第１組のシンドローム情報Ｂ１（ｉ），Ｂ２（ｉ）、及び第２組のシンドローム情報Ｃ（ｊ）で表すことができる。具体的に、図７のテキストデータを配列方向にＮ個（ｉ＝１～Ｎ）で、非配列方向にＭ個（ｊ＝１～Ｍ）の部分テキストデータＴ（ｉ，ｊ）に分割し、各部分テキストデータＴ（ｉ，ｊ）が１６個分のヌクレオチドのテキストデータを含むものとすると、元のテキストデータのデータ量ＤＴ１は、以下のようになる。

【００２８】

ＤＴ１＝１６・Ｎ・Ｍ（バイト）　　…（１）

更に、各ヌクレオチドを２ビットのデータで表すものとすると、各部分テキストデータＴ（ｉ，ｊ）は、それぞれ３２ビットの変換データＡ（ｉ，ｊ）に変換され、シンドローム情報Ｂ１（ｉ），Ｂ２（ｉ），Ｃ（ｊ）も３２ビットのデータとなる。また、非配列方向のシンドローム情報が２列Ｂ１（ｉ），Ｂ２（ｉ）あるとすると、シンドローム情報のデータ量ＤＳ１は、以下のようになる。

【００２９】

ＤＳ１＝３２（２・Ｎ＋Ｍ）（ビット）

＝４（２・Ｎ＋Ｍ）（バイト）　　…（２）

従って、仮にＮ＝６４，Ｍ＝１２８とすると、（１）式及び（２）式よりデータ量ＤＴ１，ＤＳ１は以下のようになる。

ＤＴ１＝１３１０７２（バイト）≒１３０ｋバイト　…（３）

ＤＳ１＝１０２４（バイト）＝ＤＴ１／１２８　　…（４）

従って、シンドローム情報のデータ量は、元のテキストデータのデータ量のほぼ１／１００程度に圧縮できる。この場合、例えば人間の１本の染色体のＤＮＡの配列は、５０Ｍバイト～２５０Ｍバイト程度のテキストデータで表されるため、予めそのテキストデータを５００個～２５００個程度のブロックに分割し、各ブロック毎にシンドローム情報を求めることによって、全部のシンドローム情報のデータ量はそのテキストデータのほぼ１／１００程度、即ち５００ｋバイト～２．５Ｍバイト程度に圧縮される。この程度のデータ量であれば、例えば携帯電話システムのような低速の通信回線を介しても短時間に送信できると共に、ＤＶＤ－ＲＯＭよりも容量の少ないＣＤ－ＲＯＭ等の記録媒体にも余裕を持って記録することができる。

【００３０】

この場合、複数行のその変換データの各行の変換データをそれぞれその非配列方向に交互に第１群の変換データ（例えば奇数番目の変換データＡ（ｉ，１），Ａ（ｉ，３），…）及び第２群の変換データ（例えば偶数番目の変換データＡ（ｉ，２），Ａ（ｉ，４），…）に分けたとき、その第１の演算は、所定の整数Ｋを用いてその第１群の変換データ、及びその第２群の変換データのそれぞれの法Ｋのもとの和を求める演算であり、その第２の演算は、複数列のその変換データの各列の変換データに対する法Ｋのもとの和を求める演算である。その変換データＡ（ｉ，ｊ）をｓビット（例えばｓ＝３２，ｓ＝６４等）とすると、その整数Ｋは一例として次のようになる。

【００３１】

$K=2^{s}$　　…（５）

通常のコンピュータでは、その法Ｋのもとの和演算は極めて高速に実行することができる。

また、その一列のヌクレオチドの配列を基準配列として、この基準配列の２組

のそのシンドローム情報（Ｂ１（ｉ），Ｂ２（ｉ），Ｃ（ｊ））に対応させて、検査対象の一列のヌクレオチドの配列（ＴＦ（ｉ，ｊ））の２組のシンドローム情報（Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ），ＣＦ（ｊ））を求め、その４組のシンドローム情報よりその基準配列に対するその検査対象の一列のヌクレオチドの配列の相違部を求めることが望ましい。例えば図７の配列を基準配列、図１０の配列を検査対象の配列として、図７の基準配列のシンドローム情報が図８に、図１０の配列のシンドローム情報が図１１に表されている。このとき、図８のシンドローム情報（Ｂ１（ｉ），Ｂ２（ｉ），Ｃ（ｊ））に対して、図１１のシンドローム情報（Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ），ＣＦ（ｊ））は、Ｂ１Ｆ（１），Ｂ２Ｆ（４），ＣＦ（１６），ＣＦ（１７）の値が異なるため、それらの交点として、図１０の部分テキストデータＴＦ（４，１６），ＴＦ（１，１７）が図７の基準配列と異なっていることを検出できる。即ち、４組のシンドローム情報を比較することによって、少ないデータ量の比較で、検査対象の配列のどの部分テキストデータが基準配列と異なっているかを検出できる。

【００３２】

この際に、基準配列と異なっている部分をエラーコード（error code）と呼ぶと、エラーコードが部分テキストデータの各行、又は各列に一つである場合には、それら４組のシンドローム情報、及びその検査対象のエラーコードに対応する変換データの法Ｋの加減算より、基準配列の変換データＡ（４，１６），Ａ（１，１７）、ひいては部分テキストデータＴ（４，１６），Ｔ（１，１７）が正確に復元できる。従って、例えば遺伝子中の一つの塩基（ヌクレオチド）だけが異なるＳＮＰ（一塩基変位多型：Single Nucleotide Polymorphism）は本発明によって容易に検出できると共に、それに対応する正常な配列も容易に復元できる。

【００３３】

なお、図１０の場合のように隣接する２つの列の部分テキストデータＴＦ（４，１６），ＴＦ（１，１７）に跨るような長いエラーコード（以下、「バーストエラー（burst error)」と呼ぶ）が存在する場合に、非配列方向のシンドロームが各行に１つ（即ち、Ｂ１Ｆ（ｉ）とＢ２Ｆ（ｉ）との和）のみであるとすると、１行中の２箇所の部分テキストデータ、及び１列中の２箇所の部分テキストデ

ータにエラーコードが検出されてしまう。従って、エラーコードの位置の誤検出が生じて、それに対応する基準配列の復元も困難となる。これに対して本発明のように各行で２つのシンドローム情報を求めることによって、バーストエラーの検出及び復元を正確に行うことができる。なお、各行で２つのシンドローム情報を求める代わりに、配列方向（各列）で例えば前半分と後半分との２群の変換データに対して２つのシンドローム情報を求めるようにしてもよく、どちらを採用するかは全体のデータ量が少なくなるように選択すればよい。

【００３４】

次に、本発明の記録装置は、一列のヌクレオチドの配列情報の記録装置であって、一つの核酸の少なくとも一部に含まれる一列のヌクレオチドの配列情報を読み取る配列読み取り装置（４）と、この配列読み取り装置で読み取られた配列の情報をテキストデータとして第１ファイル（１９）に記録する第１記録手段（ステップ１０２～１０４）と、その第１ファイルのテキストデータよりも少ないデータ量で、その配列読み取り装置で読み取られた配列の情報を表し、この配列の情報を第２ファイル（２０，２１）に記録する第２記録手段（ステップ１０５～１０７）とを有するものである。これによって、本発明の配列情報の記録方法が実施できる。

【００３５】

この場合、その第２記録手段は、一例としてその配列読み取り装置で読み取られた一列のヌクレオチドの配列を、この配列を表すテキストデータ又は数値データの数学的な要約値で表すものである。

また、その第２記録手段は、別の例としてその配列読み取り装置で読み取られた一列のヌクレオチドの配列に対応するテキストデータを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、その部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる６ビット以下の数値データを割り当てることによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第１の演算を施して第１組のシンドローム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第２の演算を施して第２組のシンドローム情報

を求め、その第１組及び第２組のシンドローム情報をその第２ファイルに記録するものである。

【００３６】

また、本発明の記録媒体は、一列のヌクレオチドの配列情報を記録したコンピュータ読み取り可能な記録媒体であって、その一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、その一列のヌクレオチドの配列に関する情報が記録されたものである。本発明によれば、例えば人間のＤＮＡのヌクレオチドの配列情報を、少ないデータ量で記録できるため、記録媒体としてＣＤ－ＲＯＭ，ＣＤ－Ｒ等の使い勝手の良い媒体を使用できる。

【００３７】

この場合、その一列のヌクレオチドが２５個以上のヌクレオチドの配列であるときに、その一列のヌクレオチドの配列に関する情報は、一例として４０ビット以上で１９２ビット以下の長さの数学的な要約値でその記録媒体に記録されるものである。この場合には、記録媒体としてフレキシブルディスクであっても使用できる。

【００３８】

また、別の例として、その一列のヌクレオチドの配列に対応するテキストデータを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、その部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる６ビット以下の数値データを割り当てることによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第１の演算を施して第１組のシンドローム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第２の演算を施して第２組のシンドローム情報を求めておき、その一列のヌクレオチドの配列に関する情報は、その第１組及び第２組のシンドローム情報としてその記録媒体に記録される。この記録媒体を用いることによって、例えば２つの試料のヌクレオチドの配列の相違する部分の位置の検出ができると共に、その相違する部分が少ない場合にはそれに対応する配列の復元を行うことができる。

【００３９】

次に、本発明の配列情報の供給方法は、一列のヌクレオチドの配列情報の供給方法であって、その一列のヌクレオチドの配列に対応するテキストデータ、又は複数種類のヌクレオチドに対して互いに異なる６ビット以下の数値データを割り当てることによってそのテキストデータを変換して得られる数値データを保持する供給者（２Ａ）が、その一列のヌクレオチドの配列の長さの情報、及びその配列を表すテキストデータ又はその数値データの数学的な要約値の情報を通信回線（１）を介して閲覧可能な状態にしておき、その通信回線を介してその配列の長さの情報及びその数学的な要約値の情報を閲覧したユーザ（２Ｂ）より、そのテキストデータ又はその数値データの少なくとも一部の情報に対する取得要求がその供給者に届いた後に、その供給者がそのユーザにそのテキストデータ又はその数値データの少なくとも一部の情報を供給するものである。

【００４０】

この供給方法は、上記の本発明のヌクレオチドの配列情報の記録方法を、その配列情報を供給（販売）する際のビジネスモデルに適用したものである。即ち、本発明のビジネスモデルでは、或る生物ＸのＤＮＡのヌクレオチドの配列を最初に解読した供給者は、その配列のテキストデータ（又はこれを変換した数値データ）よりハッシュ関数によって要約値（message digest）を算出し、この要約値を例えばインターネット上で閲覧可能にする。これによって、その供給者は、そのテキストデータ自体を公開することなく、最初にその生物ＸのＤＮＡの配列を解読したことを主張できる。更に、ユーザが同じ配列情報を異なる供給者から誤って購入することも防止できる。

【００４１】

また、或るユーザが、その供給者よりその生物ＸのＤＮＡの配列情報を購入した後、購入した配列情報よりそのハッシュ関数によって要約値を算出し、その配列の長さも求める。そして、この配列の長さ、及び要約値をインターネット上で公開されている値と比較することによって、購入した配列情報が正確なものであるかどうかを極めて高い確率で確認できる。

【００４２】

この場合、その一列のヌクレオチドは２５個以上のヌクレオチドの配列である

ときに、一例として、その数学的な要約値は、４０ビット以上で１９２ビット以下のデータであり、その供給者は、更にその一列のヌクレオチドの所定の一部の配列の情報をその通信回線を介して閲覧可能な状態にしておくことが望ましい。その要約値、及びその配列の長さの他に、そのように例えばその配列の先頭の８個程度、及び後端の８個程度の配列を比較することによって、同一性の確認をより高精度に行うことができる。

【００４３】

また、その供給者は、その一列のヌクレオチドの配列に対応するテキストデータ、又はこれに対応するその数値データを第１ファイル（１９）に記録して保持し、その供給者は、そのテキストデータ、又はその数値データを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分データに分割し、その部分データを、それぞれ複数種類のヌクレオチドに対して互いに異なる６ビット以下の数値データを割り当てることによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第１の演算を施して第１組のシンドローム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第２の演算を施して第２組のシンドローム情報を求め、その第１組及び第２組のシンドローム情報を第２ファイル（２０）に記録して保持し、第１段階としてそのユーザは、その供給者よりその第２ファイルに記録されている２組のシンドローム情報を受け取り、その２組のシンドローム情報に基づいて検査対象の一列のヌクレオチドの配列の内のその供給者の一列のヌクレオチドの配列との相違部を特定し、この相違部の配列の復元ができない場合に、第２段階としてそのユーザはその供給者よりその第１ファイルに記録されているそのテキストデータ、又はその数値データの内のその配列の復元ができない部分の情報の提供を要求することが望ましい。

【００４４】

このようにそのユーザが最初は、希望するヌクレオチドの配列情報のシンドローム情報のみを購入する場合には、そのデータ量の小さいシンドローム情報はその通信回線を介して短時間で受信することができる。そして、シンドローム情報だけで検査対象の配列のエラーコードの特定、及び復元ができる場合には、それ

以上の配列情報を購入する必要が無い。一方、エラーコードが多く存在し、シンドローム情報のみでは全部の正確なデータが復元できない場合には、復元できない部分のテキストデータのみを購入することによって、通信回線を介して必要な配列情報を短時間に購入できる。従って、通信回線として、携帯電話システムのような比較的低速の通信回線も使用できる。

【００４５】

次に、本発明のアミノ酸の配列情報の記録方法は、一列のアミノ酸の配列情報の記録方法であって、その一列のアミノ酸の配列に対応するテキストデータよりも少ないデータ量で、その一列のアミノ酸の配列に関する情報を記録するものである。

斯かる本発明によれば、その一列のアミノ酸は、例えば或るタンパク質を構成するアミノ酸の配列の少なくとも一部である。そのアミノ酸の配列が、そのテキストデータ以外のより少ないデータ量のファイルとして記録される。従って、記録媒体として、小容量でも通常のコンピュータで手軽に再生できる媒体を使用できると共に、通信回線を介して送信する際の時間を短縮できる。

【００４６】

本発明において、その一列のアミノ酸は、一つのタンパク質を構成する１本のアミノ酸の鎖の全部又は一部である場合に、一例としてその一列のアミノ酸の配列に対応するテキストデータが、２０種類のアミノ酸に対して互いに異なる６ビット以下のデータを割り当てることによって変換される。テキストデータ形式で一文字表記（1-Letter Code）を行うものとすると、２０種類のアミノ酸は、それぞれ８ビットのアスキーコード（ASCII CODE）、文字では例えばＡ，Ｃ，Ｅ等で表されるため、各アミノ酸を６ビット以下のデータで表すことによって、データ量を減らすことができる。

【００４７】

なお、一つのアミノ酸の種類は一列の３個のヌクレオチドの配列、即ち一つの遺伝子コドン（codon ）によって決定される。これに関して、上記のヌクレオチドの配列情報の記録方法において、各ヌクレオチドを２ビットのデータで表した場合に、１つの遺伝子コドンは６ビットのデータで表される。そこで、この各遺

伝子コドンの６ビットのデータを、対応するアミノ酸の６ビットのデータとみなしてもよい。この場合には、所定のアミノ酸を表すデータが複数存在する、即ちコードの縮重（degeneracy）が生じるため、一例として各アミノ酸のデータの内で最も小さいデータをそのアミノ酸に割り当てるようにしてもよい。これによって、ヌクレオチドとアミノ酸とで共通のコードを使用できる利点がある。また、２０種類のアミノ酸は、最も少ないデータ量では、５ビットのデータで表すことができる。

【００４８】

また、本発明において、より具体的な第１の記録方法は、その一列のアミノ酸の配列に関する情報を、その配列を表すテキストデータの数学的な要約値（message digest）で表すものである。例えば所定のハッシュ関数を用いてそのテキストデータの要約値を求め、この要約値をインターネット上で公開することによって、そのテキストデータを公開することなく、その配列を最先に解読したことを主張（証明）できる。更にそのテキストデータを購入したユーザが、購入したデータの要約値を求め、この要約値を公開されている要約値と比較することによって、購入したデータの同一性を確認できる。

【００４９】

この場合、その一列のアミノ酸は２５個以上のアミノ酸の配列であるときに、その一列のアミノ酸の配列に関する情報を１６ビット以上で１９２ビット以下の長さの数学的な要約値で表すことが望ましい。タンパク質の種類は、仮想的なものも含めて１０万（＝10^5）種類程度と言われており、次の関係が成立している。

【００５０】

$$10^5 \fallingdotseq 2^{16.6} \qquad \cdots (6)$$

従って、例えばアミノ酸の配列の個数も識別データに用いるものとすると、１６ビット以上の要約値を用いることによって、ほぼ全てのタンパク質を識別することができる。また、２５個以上のアミノ酸の配列のテキストデータは、一文字表記でも２００ビット以上になるため、１９２ビット以下の要約値のデータ量はテキストデータのデータ量よりも少なくなる。

【００５１】

また、その数学的な要約値は、その一列のアミノ酸の配列に対応するテキストデータに例えばＭＤ５ハッシュ関数（要約値は１２８ビット）、又はＳＨＳ（Secure Hash Standard）ハッシュ関数（要約値は１６０ビット）の演算を施して得られる。この場合、要約値が必要以上に長くならない点ではＭＤ５ハッシュ関数が望ましい。但し、タンパク質を構成するアミノ酸の配列の数は２０個～１０００個程度であり、要約値から元のテキストデータが推定される恐れがある。そこで、アミノ酸の配列の要約値を算出する場合で、かつ元のテキストデータの秘匿性を高めたい場合には、より複雑な演算を行って得られる要約値も長くなるＳＨＳハッシュ関数を使用することが望ましい。

【００５２】

また、本発明において、より具体的な第２の記録方法は、その一列のアミノ酸の配列に対応するテキストデータを、そのアミノ酸の配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、その部分テキストデータを、それぞれ複数種類のアミノ酸に対して互いに異なる８ビット以下の数値データを割り当てることによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第１の演算を施して第１組のシンドローム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第２の演算を施して第２組のシンドローム情報を求め、その第１組及び第２組のシンドローム情報でその一列のアミノ酸の配列を表すものである。

【００５３】

本発明は、予めそのテキストデータ（一文字表記とする）を一列の数値データに変換した後に、その数値データを複数行で複数列の変換データに分割することと実質的に等価である。本発明によれば、例えばアミノ酸の配列に対応するテキストデータを配列方向にＮ個（ｉ＝１～Ｎ）で、非配列方向にＭ個（ｊ＝１～Ｍ）の部分テキストデータＴ（ｉ，ｊ）に分割し、各部分テキストデータＴ（ｉ，ｊ）が４個分のアミノ酸のテキストデータを含むものとすると、元のテキストデータのデータ量ＤＴ２は、以下のようになる。

【００５４】

ＤＴ２＝４・Ｎ・Ｍ（バイト）　　…（７）

更に、その部分テキストデータＴ（ｉ，ｊ）をそのまま変換データＡ（ｉ，ｊ）とみなすと、変換データＡ（ｉ，ｊ）はそれぞれ３２ビットの数値データとなり、シンドローム情報も３２ビットのデータとなる。また、非配列方向のシンドローム情報が２列あるとすると、シンドローム情報のデータ量ＤＳ２は、以下のようになる。

【００５５】

ＤＳ２＝３２（２・Ｎ＋Ｍ）（ビット）

＝４（２・Ｎ＋Ｍ）（バイト）　　…（８）

従って、仮にＮ＝１６，Ｍ＝１６とすると、（７）式及び（８）式よりデータ量ＤＴ２，ＤＳ２は以下のようになる。

ＤＴ２＝１０２４（バイト）　　…（９）

ＤＳ２＝１９２（バイト）≒ＤＴ２／５．３　…（１０）

従って、シンドローム情報のデータ量は、元のテキストデータのデータ量のほぼ１／５程度に圧縮できる。個々のタンパク質の配列のテキストデータのデータ量は１ｋバイト程度以下であるが、例えば１００００種類程度のタンパク質のテキストデータをまとめると１０Ｍバイト程度になる。この際にシンドローム情報を用いることによって、通信回線を介して短時間で近似的な情報を送信することができる。また、シンドローム情報を比較することによって、標準試料のアミノ酸の配列と検査対象のアミノ酸の配列との相違部（エラーコード）を効率的に検出することができる。更に、エラーコードが各行、又は各列に１つの変換データのみであるときには、それに対応する正確な配列を復元できる。

【００５６】

特に、複数行のその変換データの各行の変換データをそれぞれその非配列方向に交互に第１群の変換データ及び第２群の変換データに分けたとき、その第１の演算は、所定の整数Ｋを用いてその第１群の変換データ、及びその第２群の変換データのそれぞれの法Ｋのもとの和を求める演算であり、その第２の演算は、複数列のその変換データの各列の変換データに対する法Ｋのもとの和を求める演算である場合には、２列に跨るような長い配列の相違（バーストエラー）であって

も正確に検出、及び復元を行うことができる。

【００５７】

次に、本発明の一列のアミノ酸の配列情報の記録装置は、一つのタンパク質の少なくとも一部に含まれる一列のアミノ酸の配列情報をテキストデータとして第１ファイルに記録する第１記録手段と、その第１ファイルのテキストデータよりも少ないデータ量で、その一列のアミノ酸の配列の情報を表し、この配列の情報を第２ファイルに記録する第２記録手段とを有するものである。この発明によって、本発明のアミノ酸の配列情報の記録方法を実施することができる。

【００５８】

この場合、そのその第２記録手段は、その一列のアミノ酸の配列を、この配列を表すテキストデータの数学的な要約値で表すことが望ましい。

また、本発明の一列のアミノ酸の配列情報の供給方法は、その一列のアミノ酸の配列に対応するテキストデータ、又は複数種類のアミノ酸に対して互いに異なる８ビット以下の数値データを割り当てることによってそのテキストデータを変換して得られる数値データを保持する供給者が、その一列のアミノ酸の配列の長さの情報、及びその配列を表すテキストデータ又はその数値データの数学的な要約値の情報を通信回線を介して閲覧可能な状態にしておき、その通信回線を介してその配列の長さの情報及びその数学的な要約値の情報を閲覧したユーザより、そのテキストデータ又はその数値データの少なくとも一部の情報に対する取得要求がその供給者に届いた後に、その供給者がそのユーザにそのテキストデータ又はその数値データの少なくとも一部の情報を供給するものである。

【００５９】

この供給方法は、上記の本発明のアミノ酸の配列情報の記録方法を、その配列情報を供給（販売）する際のビジネスモデルに適用したものである。即ち、本発明のビジネスモデルでは、或る新規のタンパク質のアミノ酸の配列を最初に決定した供給者は、その配列のテキストデータ（又はこれを変換した数値データ）よりハッシュ関数によって要約値（message digest）を算出し、この要約値を例えばインターネット上で閲覧可能にする。これによって、その供給者は、そのテキストデータ自体を公開することなく、最初にそのタンパク質の配列を決定したこ

とを主張（証明）できる。更に、ユーザが同じ配列情報を異なる供給者から誤って購入することも防止でき、競合メーカは重複投資を避けることができる。

【００６０】

また、或るユーザが、その供給者よりそのタンパク質の配列情報を購入した後、購入した配列情報よりそのハッシュ関数によって要約値を算出し、その配列の長さも求める。そして、この配列の長さ、及び要約値をインターネット上で公開されている値と比較することによって、購入した配列情報が正確なものであるかどうかを極めて高い確率で確認できる。

【００６１】

この場合、その一列のアミノ酸が２５個以上のアミノ酸の配列であるときに、その数学的な要約値は、１６ビット以上で１９２ビット以下のデータであることが望ましい。

次に、本発明の第１の要約値の計算方法は、一つ又は複数のファイルに記録されたデータの要約値を計算するための要約値の計算方法であって、その一つ又は複数のファイルに記録されたデータの内で所定のコードを無視して要約値を計算するものである。

【００６２】

本発明によれば、例えばヌクレオチドの配列を表すテキストデータの要約値を計算する場合に、その配列を見やすくするために所々にスペース、改行、及びそれまでのヌクレオチドの数を示す数字等が付加されていても、これらの付加されたコードを無視することによって、本来のヌクレオチドの配列に対応する要約値を計算することができる。

【００６３】

この場合、その無視する所定のコードの別の例は、同一又は互いに異なる２組のコード、及びこれら２組のコードに挟まれたデータである。即ち、例えばいわゆるコメント文を要約値の計算対象から除去することによって、コメント文の内容を任意に記載しても、本来のヌクレオチドの配列に対応する要約値を計算することができる。

【００６４】

この場合、そのテキストデータは、その所定のコードの他に２５個以上のヌクレオチドの配列に関するデータを含むものとすると、一例として、その要約値は、４０ビット以上で１９２ビット以下のデジタルデータである。また、その要約値を計算するための関数としては、ＭＤ５ハッシュ関数（要約値は１２８ビット）、又はＳＨＳハッシュ関数（要約値は１６０ビット）等を使用することができる。

【００６５】

また、本発明において、その一つ又は複数のファイルから１文字分のコードデータを読み出す毎に、この読み出されたコードデータがその所定のコードであるときには、この読み出されたコードデータを無視して、次の１文字分のコードデータの読み出しを行い、この読み出しによって得られたその所定のコード以外のコードデータが予め定められた個数になるか、又は読み出すべきデータがなくなったときに、要約値の計算を行うようにしてもよい。

【００６６】

このように部分的に読み出されたデータ毎にその所定のコードを無視して順次要約値の計算を行うことによって、例えば最初にその一つ又は複数のファイルからその所定のコードを取り出した新たなファイルを作成して、この新たなファイルの要約値を計算する方法と比べて、記憶装置の容量がほぼ１／２程度で済む利点がある。

【００６７】

次に、本発明の第２の要約値の計算方法は、一連のテキストデータの要約値を計算するための要約値の計算方法であって、その一連のテキストデータを先頭から順に所定個数ずつのコードデータを含む複数の部分テキストデータと、その所定個数よりも少ない個数のコードデータを含む端数のテキストデータとに分割し、その複数の部分テキストデータ、及び端数のテキストデータをそれぞれ分割する順序を含むデータとともに互いに異なる複数のファイルに記録し、この複数のファイルに記録されたテキストデータからその分割の順序に従って順次要約値を計算するものである。

【００６８】

斯かる本発明によれば、例えば人間のゲノム情報のような膨大な量のテキストデータの要約値を求める場合に、そのテキストデータを複数のファイルに分割して記録しておき、分割されたファイルのデータに順次演算処理を施すことができる。従って、ＣＤ－ＲＯＭやフレキシブルディスクのように比較的容量の少ない記録媒体を用いる場合にも、その膨大な量のテキストデータの要約値を容易に、かつ正確に計算できる。

【００６９】

この場合、そのその所定個数ずつのコードデータ、及びその所定個数よりも少ない個数のコードデータからは、所定のコードデータ（例えば数字コード、スペースコード、改行コード等）を除外してもよい。

また、その所定個数は、要約値を計算する際のデータ量の単位に応じて定めることが望ましい。例えばＭＤ５ハッシュ関数は、５１２ビット（６４バイト）のデータ単位で要約値を計算するため、一つのコードデータが８ビット（１バイト）であるとすると、その所定個数は、６４の整数倍に設定することによって、各部分テキストデータ毎の要約値の計算が容易になる。

【００７０】

【発明の実施の形態】

以下、本発明の実施の形態の一例につき図面を参照して説明する。本例は、所定のＤＮＡ（デオキシリボ核酸：deoxyribonucleic acid）のヌクレオチドの配列情報をコンピュータシステムで処理する場合に、本発明を適用したものである。

【００７１】

図１は、本例のコンピュータシステム２Ａの概略構成を示し、この図１において、コンピュータシステム２Ａの中心は、ＣＰＵ（中央演算処理ユニット）、ＲＡＭ，ＲＯＭ等のメモリ、及びハードディスク装置等の記憶装置等からなる情報処理装置１０である。情報処理装置１０には、ビデオＲＡＭ（ＶＲＡＭ）１１を介してＣＲＴディスプレイよりなる表示装置１２が接続されると共に、Ｉ／Ｏユニット（入出力装置）１４を介して、記録可能なＣＤ－Recordableディスク（以下、「ＣＤ－Ｒ」と言う）１６に対するデータの書き込み、及びＣＤ－ＲやＣＤ

－ＲＯＭからのデータの読み込みを行うことができるＣＤ－Ｒ／ＲＷドライブ１５が接続されている。情報処理装置１０には、Ｉ／Ｏユニット１４を介して更に大容量の記憶装置としての記憶容量が数１０Ｇバイト程度の磁気ディスク装置１７が接続されている。

【００７２】

本例の情報処理装置１０中のハードディスク装置には、予めＣＤ－Ｒ／ＲＷドライド１５を介してオペレーティングシステム、及び後述のようにＤＮＡの配列情報を処理するためのアプリケーション・プログラムがインストールされている。また、ＣＤ－Ｒ１６が本発明の記録媒体に対応しているが、記録媒体としては、ＣＤ－ＲやＣＤ－ＲＯＭの他に、フラッシュＲＯＭ、フレキシブルディスク、光磁気ディスク（ＭＯ）、デジタルビデオディスク（ＤＶＤ）、又はハードディスク装置（例えばインターネットを介して接続できるサーバに備えられたもの）等を使用することができる。

【００７３】

情報処理装置１０には更に、文字情報の入力装置としてのキーボード１３、ポインティング・デバイス（入力装置）としての光学式のマウス２０４、及びルータ（又はモデム等でもよい）よりなる通信制御ユニット１８が接続されている。マウス２０４は、表示装置１２の表示画面上のカーソルの位置を指定する信号を発生する変位信号発生部２０７、選択すべき情報を指定する信号や各種コマンド等を発生するための左スイッチ２０４ａ及び右スイッチ２０４ｂ（信号発生装置）を備えている。情報処理装置１０、ＶＲＡＭ１１、表示装置１２、キーボード１３、マウス２０４、Ｉ／Ｏユニット１４、ＣＤ－Ｒ／ＲＷドライブ１５、磁気ディスク装置１７、及び通信制御ユニット１８等よりコンピュータシステム２Ａが構成されている。オペレーティングシステムとして本例ではＷｉｎｄｏｗｓ（Microsoft Corporation の登録商標）を使用している。なお、オペレーティングシステムとして、それ以外のＵＮＩＸ（X/Open　の登録商標）、ＯＳ／２（IBM Corporatin　の登録商標）、ＭａｃＯＳ（Apple Computer　の登録商標）、又はＬｉｎｕｘ（Linus Torvalds　の商標又は登録商標）等を使用する場合にも本発明が適用できることは言うまでもない。

【００７４】

そして、コンピュータシステム２Ａ（情報処理装置１０）は、通信制御ユニット１８を介して一般電話回線よりなる通信ネットワーク１に接続され、通信ネットワーク１には各種コンテンツのプロバイダ３、及び別のコンピュータシステム２Ｂ、及び不図示の多くのサーバやコンピュータシステムが接続されている。また、本例のコンピュータシステム２Ａ，２Ｂ及びプロバイダ３は、通信ネットワーク１を介するインターネットによって相互に接続されている。この場合、コンピュータシステム２Ａの所有者がＤＮＡ情報の供給者（販売者）であり、コンピュータシステム２Ｂの所有者がそのＤＮＡ情報のユーザ（購入者）である。そして、後者のコンピュータシステム２Ｂには、予め前者のコンピュータシステム２Ａと同様のＤＮＡの配列情報を処理するためのアプリケーション・プログラムがインストールされている。

【００７５】

さて、本例のコンピュータシステム２Ａの情報処理装置１０には、Ｉ／Ｏユニット１４を介してＤＮＡ中の一列のヌクレオチドの配列（又は塩基の配列）を読み取るための配列読み取り装置としてのシーケンサー（DNA Sequencer)４が接続されている。シーケンサー４は、一例としてサンガーの方法（Sanger method)によってＤＮＡを構成する１対の重合体の鎖の一方の鎖のヌクレオチドの配列を読み取る。サンガーの方法は、例えば文献１（Maxim D. Frank-Kamenetskii: Unraveling DNA (the most important molecule of life, revised and updated),translated by Lev Liapin, Chapter 6(pp.59-70)(Perseus Books,1997)）に開示されている。シーケンサー４は、読み取った一列のヌクレオチドの配列をテキストデータ形式で内部の大容量の記憶装置に記憶すると共に、情報処理装置１０からの要求に応じて、その記憶装置中の所定のヌクレオチドの配列のテキストデータをＩ／Ｏユニット１４を介して情報処理装置１０に供給する。これに対して情報処理装置１０は、ＤＮＡの配列情報を処理するためのアプリケーション・プログラムに基づいて以下の処理を行う。なお、シーケンサー４の代わりに、ＤＮＡ及びＲＮＡ（リボ核酸：ribonucleic acid）等の核酸を構成する一列のヌクレオチドの配列（又は塩基の配列）の情報のデータベースを接続してもよい。

【００７６】

先ず、本例の情報処理装置１０の第１の基本的な処理動作につき説明する。情報処理装置１０は、シーケンサー４から供給される所定のＤＮＡのヌクレオチドの配列を示すテキストデータを磁気ディスク装置１７中のマスターファイル１９にそのまま記録すると共に、そのテキストデータをよりデータ量の少ない数値データに変換し、この変換後の数値データを磁気ディスク装置１７中のワーキングファイル２０に記録する。なお、以下の説明において、２進数表示の数ｋは bin(k) で、１６進数表示の数ｋは hex(k) で表すものとする。

【００７７】

この場合、ＤＮＡは４種類のヌクレオチドより構成されており、シーケンサー４から供給されるテキストデータ中では、塩基としてアデニン（adenine）を含むヌクレオチド、グアニン（guanine）を含むヌクレオチド、シトシン（cytosine）を含むヌクレオチド、及びチミン（thymine）を含むヌクレオチドがそれぞれ文字Ａ，Ｇ，Ｃ，及びＴで表されている。そして、文字Ａ，Ｇ，Ｃ，及びＴには、データ上ではそれぞれhex(41),hex(47),hex(43),hex(54)よりなる１バイト（８ビット）のアスキーデータが割り当てられている。また、ＲＮＡの場合には、チミンを含むヌクレオチドの代わりにウラシル（uracil）を含むヌクレオチドが、文字Ｕ（hex(55)）で表されている。従って、ｎ個のヌクレオチドの配列を示すテキストデータのデータ量はｎバイトとなる。なお、それらのｎ個のヌクレオチドの配列は、ｎ個の塩基（アデニン、グアニン、シトシン、チミン（又はウラシル））の配列ともみなすことができる。

【００７８】

本例ではそのテキストデータを、情報量を少なくすることなく最も少ないデータ量で表すために、ＤＮＡ中の４種類のヌクレオチドを互いに異なる２ビットのデータで表す。この際に、ＤＮＡにおいては、１対の塩基（アデニン及びチミン）が互いに相補的であり、別の１対の塩基（グアニン及びシトシン）が互いに相補的である。そこで、相補的な塩基を含む１対のヌクレオチドを互いに相補的であるとして、１対の互いに相補的なヌクレオチド、即ちアデニンを含むヌクレオチド及びチミンを含むヌクレオチドに、互いにビット反転の関係にある１対のデ

ータを割り当て、別の１対の互いに相補的なヌクレオチド、即ちグアニンを含むヌクレオチド及びシトシンを含むヌクレオチドに、互いにビット反転の関係にある別の１対のデータを割り当てる。本例ではそのデータの割り当てとして表１（変換テーブル）を用いる。なお、表１は、ヌクレオチドの配列を示すテキストデータ中の文字Ａ，Ｔ（又はＵ），Ｇ，Ｃ，をそれぞれbin(00),bin(11),bin(01),bin(10) で置換することを意味している。

【００７９】

《表１》

ヌクレオチド	２ビットのデータ
アデニンを含むヌクレオチド（Ａ）	bin(00)
チミン（ウラシル）を含むヌクレオチド（Ｔ又はＵ）	bin(11)
グアニンを含むヌクレオチド（Ｇ）	bin(01)
シトシンを含むヌクレオチド（Ｃ）	bin(10) 。

【００８０】

なお、本例では各ヌクレオチドを２ビットのデータで表しているが、これは各塩基を２ビットのデータで表すのと等価である。また、データの割り当ては表１には限定されず、例えばチミンを含むヌクレオチドをbin(00) 、アデニンを含むヌクレオチドをbin(11) とするか、又はグアニンを含むヌクレオチドをbin(10) 、シトシンを含むヌクレオチドをbin(01) としてもよい。それ以外に、アデニンを含むヌクレオチド及びチミンを含むヌクレオチドに、１対のデータbin(01),bin(10) を割り当て、グアニンを含むヌクレオチド及びシトシンを含むヌクレオチドに１対のデータbin(00),bin(11) を割り当てるようにしてもよい。また、ＲＮＡの場合には、チミンを含むヌクレオチドに割り当てられているデータをウラシルを含むヌクレオチドに割り当てて、それ以外のヌクレオチドにはＤＮＡのヌクレオチドと同じデータを割り当てればよい。

【００８１】

本例では図２に示すＤＮＡ分子５のヌクレオチドの配列情報を扱うものとする。その配列情報は、ＮＣＢＩ（The National Center for Biotechnology Information ）より提供されているウェブサイト１（ftp://ncbi.nlm.nih.gov/genbank

/genomes/bacteria/）より入手した大腸菌（Escherichia coli: E. coli）のＤＮＡの一列のヌクレオチドの配列の一部である。

【００８２】

図２において、ＤＮＡ分子５は、１対の重合体の鎖６Ａ，６Ｂ（二重らせん）より構成され、一方の重合体の鎖６Ａは、アデニンを含むヌクレオチド７Ａ、グアニンを含むヌクレオチド７Ｇ、シトシンを含むヌクレオチド７Ｃ、及びチミンを含むヌクレオチド７Ｔよりなる４種類のヌクレオチドの配列であり、他方の重合体の鎖６Ｂは、鎖６Ａに対して相補的なヌクレオチドの配列である。この際に、図１の情報処理装置１０には一方の重合体の鎖６Ａの配列を示すテキストデータ、即ち”ＡＧＣＴＴＴ・・・”の文字列のデータが供給される。それに対して、情報処理装置１０は、そのテキストデータを後述のようにＮ行でＭ列（Ｎ，Ｍは２以上の整数）のブロックに分割した後、各ブロック中の文字Ａ，Ｇ，Ｃ，Ｔを表１の変換テーブルに基づいて順次２ビットのデータに変換することによって、数値データとしてのバイナリーデータＢＮＡ（＝bin(0001101111･･･)）を得る。そして、このバイナリーデータＢＮＡが図１の磁気ディスク装置１７のワーキングファイル２０に記録される。そのバイナリーデータＢＮＡのデータ量は、元のテキストデータの１／４となっている。

【００８３】

この場合、そのワーキングファイル２０の先頭の所定数のバイトの領域に、例えばその配列がＤＮＡ又はＲＮＡのどちらかを示すデータ（即ち、bin(11) を文字Ｔ又は文字Ｕの何れに解釈するかを示すデータ）、ヌクレオチドの個数を示すデータ、及びその他の必要なデータを記録しておけばよい。また、そのワーキングファイル２０の長さが１バイト（８ビット）単位で規定されている場合に、バイナリーデータＢＮＡの末尾で１バイトの端数のデータが生じたときには、予め定めておいたダミーデータを付加すればよい。これでもデータ量は殆ど増加しない。そして、一例としてユーザ（コンピュータシステム２Ｂの所有者）から供給者（コンピュータシステム２Ａの所有者）に対して図２のＤＮＡ分子５の配列情報の購入希望が届いたときに、ワーキングファイル２０のデータが通信ネットワーク１及び不図示のプロバイダを介して、電子メールの添付ファイルとしてコン

ピュータシステム２Ｂ側に供給される。この際に、そのワーキングファイル２０のデータを更に圧縮ファイル（ＺＩＰファイル、又はＬＨＡファイル等）として送信してもよい。この際に、ワーキングファイル２０のデータ量はもとのテキストデータのほぼ１／４であるため、元のテキストデータ（更に圧縮ファイルとした場合も同様）自体を送信する場合に比べて送信時間はほぼ１／４となり、供給者側及びユーザ側双方の通信コストが低減できる。

【００８４】

そして、ユーザ側で、その受信したワーキングファイル２０のデータから図２の一方の重合体の鎖６Ａの配列のテキストデータを復元する場合には、コンピュータシステム２Ｂにおいて、ワーキングファイル２０中のバイナリーデータＢＮＡを、表１を用いて文字Ａ，Ｇ，Ｃ，Ｔ（又はＵ）の何れかに順次逆変換すればよい。また、その際に例えば図２の他方の相補的な重合体の鎖６Ｂのヌクレオチドの配列を示すテキストデータが必要になった場合には、コンピュータシステム２Ｂにおいて、図２に示すように、バイナリーデータＢＮＡのビット毎の反転操作を行って反転バイナリーデータNOT(BNA)（＝bin(1110010000･･･)）を得る。この反転バイナリーデータNOT(BNA)は、他方の重合体の鎖６Ｂのヌクレオチドの配列を示すテキストデータ（文字列”ＴＣＧＡＡＡ・・・”）を表１に従って変換したバイナリーデータＢＮＢと同一である。従って、その反転バイナリーデータNOT(BNA)を、表１を用いて文字Ａ，Ｇ，Ｃ，Ｔ（又はＵ）の何れかに順次逆変換するのみで、極めて高速に相補的な重合体の鎖６Ｂの配列のテキストデータを得ることができる。この際に、通常のコンピュータにおいては、ビット毎の反転操作は、極めて高速に実行することができる。なお、そのビット毎の反転操作は、例えばbin(111111･･･)との排他的論理和演算で代用してもよい。

【００８５】

なお、ワーキングファイル２０のデータを通信ネットワーク１を介してユーザ側に送信する代わりに、ワーキングファイル２０の内容をＣＤ－Ｒ／ＲＷドライブ１５によってＣＤ－Ｒ１６に記録し、このＣＤ－Ｒ１６を郵送等によってユーザ側に供給してもよい。例えば一人の人間の全部のＤＮＡの配列情報（ゲノム）は、テキストデータでは３Ｇバイト程度になるが、これを表１を用いて本例の数

値データとしてのバイナリーデータに変換すると、３／４Ｇバイト程度、即ち７５０Ｍバイト程度になる。現在のＣＤ－Ｒ，ＣＤ－ＲＯＭの容量は約６５０Ｍバイトであるため、その７５０Ｍバイト程度のバイナリーデータは例えば一部又は全部を圧縮ファイルとすることによって、余裕を持ってＣＤ－Ｒ１６に記録することができる。これに対して、その７５０Ｍバイト程度のデータを通信ネットワーク１を介して送信しようとすると、現状でも送信時間がかかり過ぎる場合がある。

【００８６】

また、一つのアミノ酸の種類は一列の３個のヌクレオチドの配列、即ち一つの遺伝子コドン（codon ）によって決定される。そこで、１つのアミノ酸に対応する３個のヌクレオチドをそれぞれ２ビットのデータで表したときに得られる６ビットのデータの内で、最も小さいデータでそのアミノ酸を表すものとする。具体的に、各ヌクレオチドを表１のように表した場合に、いくつかのアミノ酸について得られる６ビットの表現を以下の表２に示す。表１中で＜＞の中のデータがそのアミノ酸のデータとして使用される。これによって、ヌクレオチドとアミノ酸とで共通のコードを使用できる利点がある。

【００８７】

《表２》

アミノ酸	遺伝子コドン	６ビットのデータ
アラニン(Ala)	ＧＣＡ	＜bin(011000)＞
	ＧＣＧ	bin(011001)
	ＧＣＣ	bin(011010)
	ＧＣＵ	bin(011011)
システイン(Cys)	ＵＧＣ	＜bin(110110)＞
	ＵＧＵ	bin(110111)
グルタミン酸(Glu)	ＧＡＡ	＜bin(010000)＞
	ＧＡＧ	bin(010001)
ヒスチジン(His)	ＣＡＣ	＜bin(100010)＞
	ＣＡＵ	bin(100011)

イソロイシン(IIe)	ＡＵＡ	＜bin(001100)＞
	ＡＵＣ	bin(001110)
	ＡＵＵ	bin(001111)
リジン(Lys)	ＡＡＡ	＜bin(000000)＞
	ＡＡＧ	bin(000001)。

【００８８】

次に、本例の情報処理装置１０の第２の基本的な処理動作につき説明する。本例では、ヌクレオチドの配列を示す膨大な量のテキストデータ（又はこれを表１に基づいて変換して得られる数値データより、所定のハッシュ関数を用いて数学的な要約値（message digest）を算出する。本例ではそのハッシュ関数として、ライベスト（R. Rivest)によって提案されたＭＤ５ハッシュ関数を使用する。ＭＤ５ハッシュ関数のアルゴリズムについては、ネットワークワーキンググループ及びライベストによって開設されているウェブサイト２(http://www.kleinscmidt.com/edi/md5.htm)に開示されている。或るテキストデータ（テキストファイル）にそのＭＤ５ハッシュ関数を施すことによって、１２８ビットの要約値が得られる。通常のコンピュータでも今後は６４ビットのＣＰＵが使用されるようになると考えられるが、この場合に１２８（＝２・６４）ビットの要約値は非常に扱い易い長さである。この場合には、１９２（＝３・６４）ビットの要約値も比較的扱い易いと考えられる。

【００８９】

また、本例では、そのＭＤ５ハッシュ関数のプログラムとして、そのウェブサイト２において公開されている、ＲＳＡデータセキュリティー社（RSA Data Security Inc.）によって開発されたプログラムを使用した。

その要約値の使用方法の一例として、ＤＮＡの配列情報の供給者（情報処理装置１０）は、所定の生物のＤＮＡのヌクレオチドの配列を読み取り、これに対応するテキストデータより、上記のハッシュ関数を用いて要約値を算出し、この要約値をその生物の名称、及びＤＮＡの位置を示す情報と共にインターネット上で閲覧可能にする。これによって、その供給者は、そのテキストデータ自体を公開することなく、その生物のＤＮＡの配列情報を最先に解読したことを主張できる

と考えられる。その後、或るユーザからのその配列情報の購入希望が来たときに、その供給者は、そのヌクレオチドの配列のテキストデータを表１を用いてバイナリーデータに変換し、このバイナリーデータを例えば通信ネットワーク１を介して電子メールの形でそのユーザに送信する。これに対してユーザ側では、そのバイナリーデータを表１を用いてテキストデータに逆変換し、この逆変換されたテキストデータに上記のハッシュ関数を施して要約値を求める。

【００９０】

そして、この要約値とその供給者によって公開されている要約値とが等しいときには、購入した配列情報が、供給の保持している配列情報と等しいことが極めて高い確率で保証される。更に、ユーザ側では、複数の供給者が公開している要約値を比較することによって、同じ配列情報を異なる複数の供給者から重複して購入することを防止することができる。これらの際に、ヌクレオチドの配列の長さ、及び先端部や末尾の一部の短い配列の比較を行うことによって、その配列情報の同一性を高めることができる。

【００９１】

また、例えば所定の生物について複数の研究者が並行して研究を行っている場合に、第１の研究者が保持しているヌクレオチドの配列と第２の研究者が保持しているヌクレオチドの配列との同一性を保証する必要がある。この際に、研究対象とするＤＮＡのヌクレオチドの配列数が例えば１億個程度とすると、その配列のテキストデータは１００Ｍバイト程度になる。このような長い２つのテキストデータに対して１文字ずつの比較によって、同一性を確認するのは容易ではない。これに対して本例では、先ず第１の研究者側で、テキストデータの長さ、及びハッシュ関数による要約値を算出し、これを電子メール等で第２研究者側に送信する。これに対して、第２の研究者側でも自分のテキストデータの長さ及びハッシュ関数による要約値を算出し、これらの値を第１の研究者から送信された値と比較することによって、２つの膨大な長さのテキストデータの同一性を容易に高い確率で保証できる。この際にも、更に例えばヌクレオチドの配列の先端部及び末尾の所定長さの配列同士を比較することによって、その同一性を高めることができる。

【００９２】

なお、ハッシュ関数としては、例えば文献２(FIPS Publication 180, 1993)で開示されているように、ＮＢＳ(National Bureau of Standards)によって提案されたＳＨＳ（Secure Hash Standard）ハッシュ関数を使用してもよい。ＳＨＳハッシュ関数は、ＭＤ５ハッシュ関数よりも複雑な演算を行うと共に、１６０ビットの要約値が得られる。これに関して、例えばタンパク質を構成するアミノ酸の配列数は２０個～１０００個程度であり、特に一文字表記を使用する際にはそれに対応するテキストデータも２０バイト～１ｋバイト程度に短くなるため、要約値から元のテキストデータが推定し易いと考えられる。そこで、アミノ酸の配列情報の要約値を求める際には、ＳＨＳハッシュ関数を使用する方が望ましいことがある。

【００９３】

また、例えばヌクレオチドの配列を示す２つの膨大な長さのテキストデータの同一性を確認するために、ハッシュ関数の要約値を算出するような場合には、それ程複雑な計算を繰り返して行う必要は無いと考えられる。そこで、このような用途では、例えば文献３（R. L. Rivest:"The MD4 message digest algorithm", Lecture Notes in Computer Science, 537,303-311(1991)）で開示されているＭＤ４ハッシュ関数を使用してもよいと考えられる。また、そのように単に同一性を確認する用途では、要約値の長さも４０ビット～１２８ビット程度でよい場合がある。

【００９４】

次に、本例のＤＮＡ情報の供給者（コンピュータシステム２Ａ）と、ユーザ（コンピュータシステム２Ｂ）との間でＤＮＡの配列情報を受け渡す際のビジネスモデルの一例につき図３～図６のフローチャートを参照して詳細に説明する。先ず、ＤＮＡ情報の供給者側では、図３のステップ１０１において、シーケンサー４を使用して標準となる試料（標準試料Ｅとする）のＤＮＡ中の一方の一列のヌクレオチドの配列を読み取り、読み取った配列を表すテキストデータＴＸ１を情報処理装置１０に供給する。本例では、その標準試料Ｅを大腸菌として、そのテキストデータＴＸ１として、図７に示すように、上記のウェブサイト１から入手

した大腸菌のＤＮＡの配列情報の内の、最初から２０４８個までのヌクレオチドの配列を示すテキストデータを使用する。

【００９５】

標準試料ＥのＤＮＡ配列は配列番号１に示されている。図７のテキストデータは、配列番号１の配列から数字データを除いて、ａ，ｇ，ｃ，ｔの文字をそれぞれＡ，Ｇ，Ｃ，Ｔで置き換えたものに相当する。

次のステップ１０２において、情報処理装置１０は、供給されたテキストデータＴＸ１に上記のＭＤ５ハッシュ関数を施して１２８ビットの要約値ＡＢ１を求めると共に、そのヌクレオチドの配列の数ＮＡ１、及び先頭と末尾との８個ずつのヌクレオチドの配列ＳＴ１，ＳＢ１を求める。テキストデータＴＸ１に対する具体的な値は下記の通りである。

【００９６】

ＡＢ１＝hex(849339ac244cde42b5346ab5989aab61)　　…（１１）

ＮＡ１＝２０４８

ＳＴ１＝ＡＧＣＴＴＴＴＣ，ＳＢ１＝ＣＧＣＧＡＡＧＧ

次のステップ１０３において、情報処理装置１０は、テキストデータＴＸ１を逆方向に並べ替えたテキストデータＴＸＲ１（＝ＧＧＡＡＧＣ・・・ＴＴＴＣＧＡ）を求め、このテキストデータＴＸＲ１のＭＤ５ハッシュ関数による要約値ＡＢＲ１、及びこのテキストデータＴＸＲ１の先頭と末尾との８個ずつのヌクレオチドの配列ＳＴＲ１，ＳＢＲ１を求める。配列ＳＴＲ１，ＳＢＲ１は、上記の配列ＳＢ１，ＳＴ１をそれぞれ逆方向に並べ替えることによって容易に求めることができる。これらの具体的な値は以下の通りである。

【００９７】

ＡＢＲ１＝hex(4eb1feae30f522642b912ce3ea09652b)　　…（１２）

ＳＴＲ１＝ＧＧＡＡＧＣＧＣ，ＳＢＲ１＝ＣＴＴＴＴＣＧＡ

次のステップ１０４において、情報処理装置１０は、標準試料Ｅの名前の情報（試料を特定する情報）、配列の数ＮＡ１、テキストデータＴＸ１、配列ＳＴ１，ＳＢ１、要約値ＡＢ１、逆方向の配列ＳＴＲ１，ＳＢＲ１、及び逆方向の要約値ＡＢＲ１を磁気ディスク装置１７のマスターファイル１９に記録する。この際

に、マスターファイル１９を複数のファイルとして、テキストデータＴＸ１と、それ以外のデータとを別のファイルに記録してもよい。また、テキストデータＴＸ１が例えば１００Ｍバイト程度以上になる場合には、テキストデータＴＸ１を複数のマスターファイルに分割して記録してもよい。

【００９８】

次のステップ１０５において、情報処理装置１０は、図７に示すように、標準試料Ｅのテキストデータ ＴＸ１を配列方向（ヌクレオチドの配列方向）にＮ行で、その配列方向に直交する方向（以下、「非配列方向」という）にＭ列の１６文字の長さの部分テキストデータＴ（ｉ，ｊ）（ｉ＝１～Ｎ，ｊ＝１～Ｍ）に分割する。なお、Ｎ，Ｍはそれぞれ２以上の任意の整数であり、（１）式、（２）式を用いて既に説明したように、テキストデータＴＸ１が１００ｋバイト程度（又はこの整数倍）であるときに、このテキストデータＴＸ１に対して１／１００程度のデータ量のシンドローム情報を得たい場合には、例えばＮの値が６４、Ｍの値が１２８に設定される。以下では説明を簡単にするために、テキストデータＴＸ１を４行で、かつ３２列に分割した場合を想定する。即ち、Ｎ＝４，Ｍ＝３２とする。この場合、本例では端数は生じないが、例えば図７において、最後の部分テキストデータＴ（４，３２）中の文字が１６個より少ない場合には、足りない部分には予め定めた文字（例えば文字Ａ）をダミーデータとして付加すればよい。また、部分テキストデータＴ（ｉ，ｊ）の長さは、１６文字以外の任意の長さでよいが、処理速度を高めるためには、８文字の倍数が効率的である。

【００９９】

更に、情報処理装置１０は、図７の各部分テキストデータＴ（ｉ，ｊ）を表１（変換テーブル）に基づいてそれぞれ３２ビットのバイナリーデータ（数値データ）よりなる変換データＡ（ｉ，ｊ）に変換する。この結果、図８に示す４行で、３２列の変換データＡ（ｉ，ｊ）（１６進数表示）が得られる。また、変換データＡ（ｉ，ｊ）を対応するヌクレオチドの配列方向に連続して配列したときの集合体（数値データ）をバイナリーデータＢＮ１とする。このバイナリーデータＢＮ１は、図２の一方のバイナリーデータＢＮＡと同じものであるが、図２のバイナリーデータＢＮＡは２進数表示されており、図８のバイナリーデータ（変換

データＡ（ｉ，ｊ））は１６進数表示されている。この場合、各部分テキストデータＴ（ｉ，ｊ）の長さは１６バイト（＝１２８ビット）であるため、図７の全体のテキストデータＴＸ１に対して、図８の全体のバイナリーデータＢＮ１のデータ量は１／４に減少している。なお、図７の部分テキストデータＴ（ｉ，ｊ）と図８の変換データＡ（ｉ，ｊ）とは等価であるため、上記の方法の代わりに、元のテキストデータＴＸ１を表１に基づいてバイナリーデータＢＮ１に変換した後、このバイナリーデータＢＮ１をＮ行で、Ｍ列の変換データＡ（ｉ，ｊ）に分割してもよい。

【０１００】

次のステップ１０６において、情報処理装置１０は、図８の全部の変換データＡ（ｉ，ｊ）の内で、各列の変換データＡ（ｉ，ｊ）の配列方向に対する法2^{32}（ｍｏｄ2^{32}）のもとでの和、即ち配列方向のシンドローム（syndrome）Ｃ（ｊ）（ｊ＝１～３２）を計算する。Ｃ（ｊ）は以下のように表すことができる。

$$C(j)=A(1,j)+A(2,j)+\cdots+A(4,j)\ (\mathrm{mod}2^{32}) \qquad \cdots(13)$$

更に、情報処理装置１０は、各行の変換データＡ（ｉ，ｊ）（ｉ＝１～４）の内で奇数番目の変換データＡ（ｉ，２ｊ’－１）（ｊ’＝１～１６）の非配列方向に対する法2^{32}のもとでの和、及び偶数番目の変換データＡ（ｉ，２ｊ’）の非配列方向に対する法2^{32}のもとでの和、即ち非配列方向のシンドロームＢ１（ｉ），Ｂ２（ｉ）（ｉ＝１～４）を次式より計算する。

【０１０１】

$$B1(i)=A(i,1)+A(i,3)+\cdots+A(i,31)\ (\mathrm{mod}2^{32}) \qquad \cdots(14)$$

$$B2(i)=A(i,2)+A(i,4)+\cdots+A(i,32)\ (\mathrm{mod}2^{32}) \qquad \cdots(15)$$

変換データＡ（ｉ，ｊ）に対する実際の計算結果が、図８のシンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）として表示されている。

また、図９は、図８の標準試料Ｅのデータ中からシンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）だけを取り出して表示したものである。この例においては、シンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）はそれぞれ３２ビット（４バイト）であるため、全部のシンドロームのデータ量は、１６０（＝４・４０）バイトとなる。従って、全部のシンドロームのデータ量は、図７の全体のテキストデ

ータＴＸ１（２０４８バイト）に対してほぼ１／１３に減少しており、図８の全体のバイナリーデータＢＮ１に対してもほぼ１／３に減少している。

【０１０２】

次に図４のステップ１０７において、情報処理装置１０は、標準試料Ｅの名前の情報、配列の数ＮＡ１、バイナリーデータＢＮ１、シンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）を磁気ディスク装置１７のワーキングファイル２０に記録する。この際に、ワーキングファイル２０を複数のファイルとして、バイナリーデータＢＮ１と、シンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）とを別のファイルに記録してもよい。更に、バイナリーデータＢＮ１と共に、ステップ１０２で算出した要約値ＡＢ１をワーキングファイル２０に記録してもよい。

【０１０３】

また、バイナリーデータＢＮ１が長いときには、バイナリーデータＢＮ１を複数のファイルに分割して記録してもよい。更に、図７のテキストデータＴＸ１（ひいては図８のバイナリーデータＢＮ１）がかなり長い場合には、テキストデータＴＸ１を例えば１００ｋバイト程度を単位として複数のデータ群に分割し、各データ群毎にシンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）を求めるようにしてもよい。

【０１０４】

更に、ステップ１０７において、ＤＮＡ情報の供給者は、ワーキングファイル２０に記録した情報、即ち標準試料Ｅの名前の情報、配列の数ＮＡ１、バイナリーデータＢＮ１、シンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）と、マスターファイル１７に記録した要約値ＡＢ１，ＡＢＲ１の情報とを、ＣＤ－Ｒ／ＲＷドライブ１５を介してＣＤ－Ｒ１６に記録してもよい。このＣＤ－Ｒ１６から、更に多数のＣＤ－ＲＯＭを作製してもよく、これらの記録媒体が郵送等によってユーザに販売される。

【０１０５】

次の、ステップ１０８において、情報処理装置１０は、標準試料Ｅの名前の情報、配列の数ＮＡ１、配列ＳＴ１，ＳＢ１、要約値ＡＢ１、逆方向の配列ＳＴＲ１，ＳＢＲ１、及び逆方向の要約値ＡＢＲ１を磁気ディスク装置１７のコンテン

ツファイル２１に記録する。図７のテキストデータＴＸ１が仮に１００Ｍバイト程度の膨大なものであっても、コンテンツファイル２１に記録されるデータは５００バイト程度の僅かなものである。更に、情報処理装置１０は、コンテンツファイル２１中の情報を通信ネットワーク１を介してコンテンツのプロバイダ３に送信する。これによって、コンテンツファイル２１中の情報はプロバイダ３のサーバ内の閲覧可能なコンテンツファイル３１に記録されて、第３者がインターネットを介して自由に閲覧できるようになる。

【０１０６】

次のステップ１０９において、ＤＮＡ情報の供給者は、ユーザから購入要求が来るのを待つ状態となる。そして、（ａ）ユーザから標準試料Ｅに対する簡易データの要求があったときには、ステップ１１０に移行して、情報処理装置１０は、磁気ディスク装置１７のワーキングファイル２０の中のシンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）の情報を例えば電子メールの添付ファイルとしてそのユーザに送信する。一方、ステップ１０９において、（ｂ）ユーザから完全データの要求があったときには、ステップ１１１に移行して、情報処理装置１０は、ワーキングファイル２０中のバイナリーデータＢＮ１をＺＩＰファイル等の形式で圧縮し、この圧縮されたデータを例えば電子メールの添付ファイルとしてそのユーザに送信する。この際に必要に応じて、ハッシュ関数による要約値ＡＢ１を同時に送信してもよい。本例によれば、簡易データ（シンドローム）はデータ量が少ないために短時間で送信することができる。また、完全データ（バイナリーデータＢＮ１）でも元のテキストデータに比べて１／４のデータ量であるため、比較的短時間に送信することができる。

【０１０７】

また、ステップ１０９において、ユーザは、必要に応じて部分データ、即ち図８の全部の変換データＡ（ｉ，ｊ）の内の所望のデータ、例えば２つの変換データＡ（４，１６）及びＡ（１，１７）のみをその供給者から購入するようにしてもよい。これによって、必要な正確なデータのみを短時間に入手することができる。

【０１０８】

次に、ＤＮＡ情報のユーザ（図１のコンピュータシステム２Ｂの所有者とする）側では、図５のステップ１２１において、図１の通信ネットワーク１（インターネット）を介してプロバイダ３のサーバ内のコンテンツファイル３１の内容を閲覧し、その中からステップ１０８で供給者から送信された情報、即ち標準試料Ｅの名前の情報、ヌクレオチドの配列の数ＮＡ１、配列ＳＴ１，ＳＢ１、要約値ＡＢ１、逆方向の配列ＳＴＲ１，ＳＢＲ１、及び逆方向の要約値ＡＢＲ１を読み取り、読み取った情報をコンピュータシステム２Ｂ内の記憶装置の一時ファイルに記録する。

【０１０９】

次の、ステップ１２２において、そのユーザは、不図示のＤＮＡのシーケンサーを用いて、標準試料Ｅと同じ種類で検査対象の試料ＦのＤＮＡ中の一方の一列のヌクレオチドの配列を読み取り、読み取られた配列を示すテキストデータＴＸ２をコンピュータシステム２Ｂ内の情報処理装置に取り込む。その検査対象の試料Ｆとは、例えば突然変異を起こしていると思われる大腸菌であり、そのテキストデータＴＸ２は、標準試料ＥのテキストデータＴＸ１と同様に最初から２０４８個までのヌクレオチドの配列を示すものとする。

【０１１０】

試料ＦのＤＮＡ配列は配列番号２に示されている。後述の図１０のテキストデータは、配列番号２の配列から数字データを除いて、ａ，ｇ，ｃ，ｔの文字をそれぞれＡ，Ｇ，Ｃ，Ｔで置き換えたものに相当する。

図１０は、その試料ＦのＤＮＡのヌクレオチドの配列に対応するテキストデータＴＸ２を示し、この図１０の配列の内のアンダーラインを付した部分のみが、図７の標準試料Ｅの配列と異なっている。即ち、試料Ｆの配列は、標準試料Ｅの部分テキストデータＴ（４，１６），Ｔ（１，１７）の部分だけが以下のように異なっている。なお、この段階では、ユーザは、試料Ｆの配列と標準試料Ｅの配列とのどの部分が相違しているのかは分からない。

【０１１１】

標準試料Ｅ		試料Ｆ
T(4,16)=ATTTGGACGGACGTTG	→	ATTTGGAC<u>ATTATGGC</u>

T(1,17)=ACGGGGTCTATACCTG → GGCCAACTTATACCTG

そして、ユーザのコンピュータシステム２Ｂ側の情報処理装置においても、ＤＮＡの配列情報を処理するためのアプリケーション・プログラムが起動されている。そして、その情報処理装置は、ステップ１２３において、読み取られたテキストデータＴＸ２に上記のＭＤ５ハッシュ関数を施して１２８ビットの要約値ＡＢ２を求めると共に、そのヌクレオチドの配列の数ＮＡ２、及び先頭と末尾との８個ずつのヌクレオチドの配列ＳＴ２，ＳＢ２を求め、これらを内部の記憶装置の第１データファイルに記録する。テキストデータＴＸ２（図１０）に対する具体的な値は下記の通りである。

【０１１２】

ＡＢ２＝hex(1457b51222a83c3222e87cb4d4e63305)　　…（１６）

ＮＡ２＝２０４８

ＳＴ２＝ＡＧＣＴＴＴＴＣ，ＳＢ２＝ＣＧＣＧＡＡＧＧ

次のステップ１２４において、情報処理装置は、試料Ｆの配列数ＮＡ２と標準試料Ｅの配列数ＮＡ１とが等しいかどうかを調べ、両者が異なっている場合には、ユーザはステップ１２５に移行して、別のＤＮＡ情報を検索し、ＮＡ２と同じ配列数のＤＮＡ情報をサーチする。本例では、ステップ１２４において、ＮＡ２＝ＮＡ１であるため、動作はステップ１２６に移行して、試料Ｆの先頭と末尾との一部の配列ＳＴ２，ＳＢ２が、標準試料Ｅの配列ＳＴ１，ＳＢ１と等しいかどうか、更に試料Ｆの要約値ＡＢ２が標準試料Ｅの要約値ＡＢ１（ステップ１２１で一時ファイルに記録されている）と等しいかどうかを調べる。これらが共に等しい場合には、試料Ｆの配列と標準試料Ｅの配列とは非常に高い確率（ほぼ１／2^{128} ≒１／10^{38}程度の確率）で一致しているとみなすことができる。従って、ステップ１２７に移行して、コンピュータシステム２Ｂの情報処理装置は、その第１データファイルに「試料ＦのＤＮＡ構造は、標準試料ＥのＤＮＡ構造と同一である。」との情報を記録する。

【０１１３】

但し、本例では、ＳＴ２＝ＳＴ１，ＳＢ２＝ＳＢ１が成立するが、（１１）式及び（１６）式よりＡＢ２≠ＡＢ１であるため、動作はステップ１２６からステ

ップ１２８に移行して、その情報処理装置は、試料Ｆの先頭と末尾との一部の配列ＳＴ２，ＳＢ２が、標準試料Ｅを逆に並べた配列の一部の配列ＳＴＲ１，ＳＢＲ１と等しいかどうか、更に試料Ｆの要約値ＡＢ２が標準試料Ｅを逆に並べた配列の要約値ＡＢＲ１と等しいかどうかを調べる。これらが共に等しい場合には、試料Ｆの配列と標準試料Ｅを逆に並べた配列とは非常に高い確率で一致しているとみなすことができる。従って、ステップ１３９に移行して、コンピュータシステム２Ｂの情報処理装置は、その第１データファイルに「試料ＦのＤＮＡ構造は、標準試料ＥのＤＮＡ構造に対して回文（palindrome）の関係にある。」との情報を記録する。

【０１１４】

本例では、ＳＴ２≠ＳＴＲ２，ＳＢ２≠ＳＢＲ２、かつ（１２）式及び（１６）式よりＡＢ２≠ＡＢＲ１であるため、動作はステップ１２８からステップ１２９に移行して、そのユーザは、通信ネットワーク１（インターネット）を介してＤＮＡ情報の供給者から上記の簡易データ、即ち標準試料ＥのシンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）の情報（図９の情報）を購入し、購入した情報をコンピュータシステム２Ｂ（情報処理装置）内の記憶装置の第２データファイルに記録する。

【０１１５】

次に、図６のステップ１３０において、コンピュータシステム２Ｂの情報処理装置は、図１０に示すように、試料Ｆのテキストデータ ＴＸ２を配列方向（ヌクレオチドの配列方向）にＮ行で、非配列方向にＭ列の１６文字の長さの部分テキストデータＴＦ（ｉ，ｊ）（ｉ＝１～Ｎ，ｊ＝１～Ｍ）に分割する。分割数Ｎ，Ｍは標準試料Ｅの分割数と同じであり、本例では、Ｎ＝４，Ｍ＝３２である。更に、情報処理装置は、図１０の各部分テキストデータＴＦ（ｉ，ｊ）を表１（変換テーブル）に基づいてそれぞれ３２ビットのバイナリーデータ（数値データ）よりなる変換データＡＦ（ｉ，ｊ）に変換する。この結果、図１１に示す４行で、３２列の変換データＡＦ（ｉ，ｊ）（１６進数表示）が得られる。また、変換データＡＦ（ｉ，ｊ）を連続して配列した集合体（数値データ）をバイナリーデータＢＮ２とする。

【０１１６】

次に、情報処理装置は、ステップ１０６の動作と同様にして、図１１の全部の変換データＡＦ（ｉ，ｊ）の内で、各列の変換データＡＦ（ｉ，ｊ）の配列方向に対する法2^{32}（ｍｏｄ2^{32}）のもとでの和、即ち配列方向のシンドロームＣＦ（ｊ）（ｊ＝１～３２）を計算する。ＣＦ（ｊ）は、（１３）式でＡ（ｉ，ｊ）をＡＦ（ｉ，ｊ）で置き換えた式で計算される。更に、情報処理装置は、各行の変換データＡＦ（ｉ，ｊ）（ｉ＝１～４）の内で奇数番目の変換データＡＦ（ｉ，２ｊ’－１）（ｊ’＝１～１６）の非配列方向に対する法2^{32}のもとでの和、及び偶数番目の変換データＡＦ（ｉ，２ｊ’）の非配列方向に対する法2^{32}のもとでの和、即ち非配列方向のシンドロームＢ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）（ｉ＝１～４）を計算する。Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）は、（１４）式、（１５）式でＡ（ｉ，ｊ）をＡＦ（ｉ，ｊ）で置き換えた式で計算される。変換データＡＦ（ｉ，ｊ）に対する実際の計算結果が、図１１のシンドロームＣＦ（ｊ），Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）として表示されている。

【０１１７】

図８（標準試料Ｅ）と図１１（試料Ｆ）とを比較すると、図８の変換データＡ（４，１６），Ａ（１，１７）に対して図１１の変換データＡＦ（４，１６），ＡＦ（１，１７）の値が異なっている。従って、それに対応して図１１にアンダーラインを付して示すように、図１１の配列方向の２つのシンドロームＣＦ（１６），ＣＦ（１７）、及び非配列方向の２つのシンドロームＢ１Ｆ（１），Ｂ２Ｆ（４）の値が、図８の対応するシンドロームＣ（１６），Ｃ（１７），Ｂ１（１），Ｂ２（４）の値と異なっている。

【０１１８】

また、図１２は、主に図１１の試料Ｆのデータ中からシンドロームＣＦ（ｊ），Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）だけを取り出して表示したものである。

次に、ステップ１３１において、その情報処理装置は、供給者から購入した簡易データの１組のシンドローム、即ち図８（標準試料Ｅ）の１組のシンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）と、上記のように求めた試料Ｆの１組のシンドロームＣＦ（ｊ），Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）とを比較して、相違するシン

ドロームをサーチする。本例では、図１１のシンドロームＣＦ（１６），ＣＦ（１７）、及びシンドロームＢ１Ｆ（１），Ｂ２Ｆ（４）が相違するシンドロームとして特定される。この場合、配列方向の相違するシンドロームＣＦ（１６），ＣＦ（１７）の列と、非配列方向の相違するシンドロームＢ１Ｆ（１），Ｂ２Ｆ（４）の行との交点が、標準試料Ｅに対して相違する変換データの位置となる。従って、図１１の第４行で第１６列の変換データＡＦ（４，１６）、及び第１行で第１７列の変換データＡＦ（１，１７）が相違する変換データとして特定される。

【０１１９】

次のステップ１３２において、その情報処理装置は、図１１の変換データＡＦ（ｉ，ｊ）中で図８の変換データＡ（ｉ，ｊ）と相違する変換データ（ＡＦ（ｉ’，ｊ’）とする）は、各行、又は各列に多くとも一つかどうかを調べる。これが成立する場合には、その変換データＡＦ（ｉ’，ｊ’）に対応する標準試料Ｅの変換データは、法2^{32}のもとでの連立方程式によって容易に求めることができる。本例では、それが成立する、即ち相違する変換データは、第１行、第４行に一つずつで、かつ第１６列、第１７列に一つずつであるため、動作はステップ１３３に移行する。そして、その情報処理装置は、先ず変換データＡＦ（４，１６）から標準試料Ｅの変換データＡ（４，１６）を復元するために、図８のシンドロームＣ（１６）、図１１のシンドロームＣＦ（１６）、及び変換データＡＦ（４，１６）を用いて次の演算を行う。

【０１２０】

$$A(4,16)=C(16)-CF(16)+AF(4,16)\ (\mathrm{mod}2^{32})$$
$$=\mathrm{hex}(7c33894d)-\mathrm{hex}(7c3373a6)+\mathrm{hex}(3f523cd6)\ (\mathrm{mod}2^{32})$$
$$=\mathrm{hex}(3f52527d) \quad \cdots (17)$$

この結果を図８の変換データＡ（４，１６）と比較すると、復元が正確に行われていることが分かる。

【０１２１】

続いて、情報処理装置は、変換データＡＦ（１，１７）から標準試料Ｅの変換データＡ（１，１７）を復元するために、図８のシンドロームＣ（１７）、図１

１のシンドロームＣＦ（１７）、及び変換データＡＦ（１，１７）を用いて次の演算を行う。

Ａ（１，１７）＝C(17) －CF(17)＋AF(1,17) (mod2^{32})

＝hex(31b4c2ad) －hex(6661c2ad) ＋hex(5a0bccad) (mod2^{32})

＝hex(255eccad)　　…（１８）

この結果を図８の変換データＡ（１，１７）と比較すると、復元が正確に行われていることが分かる。また、復元された変換データＡ（４，１６），Ａ（１，１７）が、図１２中の試料Ｆのシンドローム ＣＦ（ｊ），Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）の内側に表示されている。図１２の変換データＡ（４，１６），Ａ（１，１７）を表１に従って逆変換して得られる部分テキストデータは、図７の標準試料Ｅの部分テキストデータＴ（４，１６），Ｔ（１，１７）と等しいことが分かる。

【０１２２】

次のステップ１３４において、その情報処理装置は、復元された変換データＡ（ｉ’，ｊ’）、即ちＡ（４，１６），Ａ（１，１７）で、図１１の試料Ｆのバイナリーデータ ＢＮ２中の対応する変換データＡＦ（４，１６），ＡＦ（１，１７）を置き換えた後、この置き換えによって得られるバイナリーデータＢＮ２を表１に基づいてテキストデータＴＸ１’に逆変換する。更に情報処理装置は、そのテキストデータＴＸ１’よりＭＤ５ハッシュ関数を用いて１２８ビットの要約値ＡＢ１’を算出し、この要約値ＡＢ１’が標準試料Ｅの要約値ＡＢ１（ステップ１２１で一時ファイルに記録されている）と等しいかどうかを確認する。本例では、ＡＢ１’＝ＡＢ１が成立するが、例えば図１１の試料Ｆの変換データＡＦ（ｉ，ｊ）中の相違する変換データの位置や内容によっては、その相違する位置がステップ１３２で正確に検出されない可能性がある。このような場合に、ＡＢ１’≠ＡＢ１となったときには、ステップ１３５に移行すればよい。通常は、ＡＢ１’＝ＡＢ１が成立して、動作はステップ１３８に移行して、情報処理装置は、上記の第１データファイルに「試料Ｆの配列と標準試料Ｅの配列との内で相違する部分の位置（ｉ’，ｊ’）、及び相違する部分テキストデータの対」の情報を記録する。本例では、位置（ｉ’，ｊ’）として位置（４，１６），（１，１

７）が、相違する部分テキストデータの対としてＡ（４，１６），ＡＦ（４，１６）及びＡ（１，１７），ＡＦ（１，１７）が記録される。

【０１２３】

一方、ステップ１３２において、相違する変換データＡＦ（ｉ’，ｊ’）が少なくとも一行に２個以上で、かつ列方向にも２個以上（奇数番目又は偶数番目で２個以上を意味する）存在する場合には、変換データの正確な復元は困難である。そこで、動作はステップ１３５に移行して、そのユーザはそのＤＮＡ情報の供給者から標準試料Ｅの完全データ、即ち図８のバイナリーデータＢＮ１を通信ネットワーク１（インターネット）を介して購入し、コンピュータシステム２Ｂの情報処理装置は、そのバイナリーデータＢＮ１を記憶装置の第３データファイルに記録する。

【０１２４】

次のステップ１３６において、その情報処理装置は、そのバイナリーデータＢＮ１を表１に基づいてテキストデータＴＸ１’に逆変換（復元）し、そのテキストデータＴＸ１’よりＭＤ５ハッシュ関数を用いて１２８ビットの要約値ＡＢ１’を算出し、この要約値ＡＢ１’が標準試料Ｅの要約値ＡＢ１（ステップ１２１で一時ファイルに記録されている）と等しいかどうかを確認する。通常は、ＡＢ１’＝ＡＢ１が成立するが、例えば通信エラー等によって送信されたバイナリーデータＢＮ１の中にエラーが生じている場合には、ＡＢ１’≠ＡＢ１となる。このときには、例えば供給者に完全データの再送信を行う等の対処を行う。そして、ステップ１３６でＡＢ１’＝ＡＢ１が成立するときには、ステップ１３７に移行して情報処理装置は、標準試料Ｅのバイナリーデータ ＢＮ１中で、試料Ｆの相違している変換データＡＦ（ｉ’，ｊ’）に対応する変換データＡ（ｉ’，ｊ’）を求める。その後、動作はステップ１３８に移行する。

【０１２５】

このように本例のビジネスモデルによれば、第１段階として標準試料Ｅの要約値ＡＢ１と試料Ｆの要約値ＡＢ２とを比較して、両者が等しいときには試料ＦのＤＮＡの構造は標準試料ＥのＤＮＡの構造と同じとみなすため、ＤＮＡ情報の供給者からそれ以上の情報を購入する必要が無い。また、第２段階として、標準試

料Ｅのシンドローム Ｃ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）と試料Ｆのシンドローム ＣＦ（ｊ），Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）とを比較して、相違する変換データＡＦ（ｉ，ｊ）の個数が少ない場合には、対応する標準試料Ｅの変換データＡ（ｉ，ｊ）を復元するため、膨大な完全データを購入する必要がなく、情報処理コストを低減できる。

【０１２６】

なお、上記のステップ１３５では、ユーザはＤＮＡ情報の供給者から完全データ（バイナリーデータＢＮ１）を購入しているが、別の方法として、ステップ１３１で特定された相違する変換データＡＦ（ｉ’，ｊ’）に対応する標準試料Ｅの変換データＡ（ｉ’，ｊ’）のみを購入してもよい。これによって、通信コストを低減できる。

【０１２７】

また、本例のシンドロームの使用方法に関して、本例では非配列方向（列方向）に２組のシンドロームＢ１（ｉ），Ｂ２（ｉ）を求めているため、図１１の試料Ｆの変換データＡＦ（ｉ，ｊ）において、連続する２列の変換データＡＦ（ｉ，ｊ）、例えばＡＦ（ｉ，１６），ＡＦ（ｉ，１７）（ｉ＝１～４）の全部（８個）が標準試料Ｅの変換データＡ（ｉ，ｊ）と相違していても、その相違する部分（エラーコード）の位置を正確に検出することができる。更に、非配列方向のシンドロームＢ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）、及び変換データＡＦ（ｉ，１６），ＡＦ（ｉ，１７）（ｉ＝１～４）を用いて連立方程式を解くことによって、対応する標準試料Ｅの変換データＡ（ｉ，ｊ）の全部を正確に復元できる。即ち、ヌクレオチドの配列方向に対して隣接する２列に跨るような比較的長いエラーコード（バーストエラー）が生じても、本例のシンドロームによってその位置の検出、及び対応する配列の復元を行うことができる。

【０１２８】

また、本例のシンドロームを用いれば、ＳＮＰ（一塩基変位多型：Single Nucleotide Polymorphism）のように所定の範囲内で１つのヌクレオチド（塩基）だけが異なっているようなエラーコードは、更に容易にその位置の検出、及び復元を行うことができる。そして、所定の範囲内で、即ち図１１の配列中で１つ（ヌ

クレオチドの１つ分）だけ生じたエラーコードの検出、及び復元を行えば良い場合には、非配列方向のシンドロームＢ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）の代わりに、それらの和（ＢＦ（ｉ）とする）を使用するのみで十分である。この場合には、図８の標準試料Ｅについても、非配列方向のシンドロームＢ１（ｉ），Ｂ２（ｉ）の代わりに、それらの和（Ｂ（ｉ）とする）を用意するのみでよい。

【０１２９】

また、例えば図８（図１１でも同様）において、配列方向のシンドロームＣ（ｊ）を、各列で前半の１組の変換データと後半の１組の変換データとで２つ設け、非配列方向のシンドロームＢ１（ｉ），Ｂ２（ｉ）を一つＢ（ｉ）とした場合にも、上記の隣接する２列に跨るバーストエラーの検出及び復元を行うことができる。また、より多くのエラーコードの復元を行うためには、計算は極めて複雑になるが、シンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）の代わりに、シンドローム情報として例えばリードソロモンのＣＲＣ符号（Reed-Solomon Cyclic Redundancy Check Code: RSCRC Code)を使用してもよい。ＲＳＣＲＣ符号については、例えば文献（James S. Plank: Software-Practice & Experience, 27(9), September, pp.995-1012(1997))で開示されている。

【０１３０】

なお、上記の実施の形態では、ＤＮＡ又はＲＮＡを構成するヌクレオチドは４種類であるため、テキストデータＴＸ１をバイナリーデータＢＮ１に変換する際に、表１に示すように各ヌクレオチドを２ビットのデータで表している。これに対して、ヌクレオチド（又は塩基）を表すテキストデータとして、以下のような１６種類の文字ａ～ｎ（８ビットのアスキーデータ）が使用されることがある。

【０１３１】

ａ　アデニン（アデニンを含むヌクレオチドと同義、以下同様）

ｃ　シトシン

ｇ　グアニン

ｔ　チミン

ｕ　ウラシル

ｍ　アデニン、又はシトシン

ｒ　　グアニン、又はアデニン

ｗ　　アデニン、又はチミン（若しくはウラシル）

ｓ　　グアニン、又はシトシン

ｙ　　チミン（若しくはウラシル）、又はシトシン

ｋ　　グアニン、又はチミン（若しくはウラシル）

ｖ　　アデニン、グアニン、又はシトシン

ｈ　　アデニン、シトシン、又はチミン（若しくはウラシル）

ｄ　　アデニン、グアニン、又はチミン（若しくはウラシル）

ｂ　　グアニン、シトシン、又はチミン（若しくはウラシル）

ｎ　　（アデニン、シトシン、グアニン、又はチミン（若しくはウラシル））又は（不明若しくは他の塩基）。

【０１３２】

この場合には、これら１６種類の文字を互いに異なる４ビットのコードに変換し、このコードを用いてテキストデータを数値データ（バイナリーデータ）に変換してもよい。これによって、データ量を１／２にすることができる。また、将来的にヌクレオチド（塩基）の種類が増加したような場合には、これらのヌクレオチドを５ビット、又は６ビットのデータで表現するようにしてもよい。

【０１３３】

また、上記の実施の形態では、図７及び図１０のヌクレオチドの配列を示すテキストデータよりハッシュ関数によって要約値を算出しているが、情報量としては、それらのテキストデータは図８及び図１１のバイナリーデータ（数値データ）と等価である。従って、これらのバイナリーデータよりハッシュ関数によってそれぞれ要約値を算出し、これらの算出結果同士を比較するようにしてもよい。バイナリーデータはテキストデータに対して１／４程度であるため、要約値を算出する時間が短縮できる利点がある。

【０１３４】

なお、上記の実施の形態では、ＤＮＡ又はＲＮＡ中のヌクレオチドの配列（又は塩基の配列）の情報を処理対象としているが、本発明は、遺伝子を形成するヌクレオチドの配列の情報を処理する場合にも適用できることは言うまでもない。

次に、本発明の実施の形態の他の例につき説明する。本例は、タンパク質又はペプチドを構成するアミノ酸の配列情報を処理する場合に本発明を適用したものである。

【０１３５】

本例でも基本的に図１のコンピュータシステム２Ａを使用できるが、ＤＮＡのシーケンサー４の代わりに、タンパク質のアミノ酸の配列を決定する配列読み取り装置としてのタンパク質用のシーケンサー（protein Sequencer)が情報処理装置１０に接続される点が異なっている。なお、その配列読み取り装置としては、アミノ酸の配列のデータベースも使用できる。本例でも、例えば新規の試料Ｇのタンパク質のアミノ酸の配列をそのシーケンサーで解読した場合に、その配列を示すテキストデータ（ＴＸ３とする）が情報処理装置１０に供給される。本例では一文字表記を採用するものとして、ｎ個のアミノ酸の配列に対応するテキストデータは、ｎバイトの長さである。本例では、その試料Ｇを大腸菌として、そのテキストデータＴＸ３として、図１３に示すように、上記のウェブサイト１から入手した大腸菌の或るタンパク質の８２０個のアミノ酸の配列を示すテキストデータを使用する。

【０１３６】

試料Ｇのアミノ酸配列は配列番号３に示されている。図１３のテキストデータは、配列番号３の配列から数字データを除いて、その配列を一文字表記で表したものに相当する。また、図１３においては、そのテキストデータが配列方向（アミノ酸の配列方向）に８行で、その配列方向に直交する非配列方向に２６列の４文字の長さの部分テキストデータに分割されており、８６１番以上のアミノ酸を示すデータの位置には０が表示されている。

【０１３７】

次に、情報処理装置１０は、供給されたテキストデータＴＸ３に上記のＭＤ５ハッシュ関数を施して１２８ビットの要約値ＡＢ３を求めると共に、そのアミノ酸の配列の数ＮＡ３、及び先頭と末尾との８個ずつのアミノ酸の配列ＳＴ３，ＳＢ３を求める。テキストデータＴＸ３に対する具体的な値は下記の通りである。

ＡＢ３＝hex(0f66dc2b3024a9739d0e912fde12b8ba)　　…（１９）

ＮＡ３＝８２０

ＳＴ３＝ＭＲＶＬＫＦＧＧ，ＳＢ３＝ＴＬＳＷＫＬＧＶ

次に、情報処理装置１０は、テキストデータＴＸ３を逆方向に並べ替えたテキストデータＴＸＲ３（＝ＶＧＬＫＷＳ・・・・ＦＫＬＶＲＭ）を求め、このテキストデータＴＸＲ３のＭＤ５ハッシュ関数による要約値ＡＢＲ３、及びこのテキストデータＴＸＲ３の先頭と末尾との８個ずつのアミノ酸の配列ＳＴＲ３，ＳＢＲ３を求める。配列ＳＴＲ３，ＳＢＲ３は、上記の配列ＳＢ３，ＳＴ３をそれぞれ逆方向に並べ替えることによって容易に求めることができる。これらの具体的な値は以下の通りである。テキストデータＴＸＲ３の配列は、テキストデータＴＸ３の配列に対して回文（palindrome）の関係にあると言うことができる。

【０１３８】

ＡＢＲ３＝hex(e895f433e1e77f84b3cadeead1a52380)　…（２０）

ＳＴＲ３＝ＶＧＬＫＷＳＬＴ，ＳＢＲ３＝ＧＧＦＫＬＶＲＭ

次に、情報処理装置１０は、試料Ｇの名前の情報（試料を特定する情報）、配列の数ＮＡ３、テキストデータＴＸ３、配列ＳＴ３，ＳＢ３、要約値ＡＢ３、逆方向の配列ＳＴＲ３，ＳＢＲ３、及び逆方向の要約値ＡＢＲ３を磁気ディスク装置１７のマスターファイル１９に記録する。この際に、マスターファイル１９を複数のファイルとして、テキストデータＴＸ３と、それ以外のデータとを別のファイルに記録してもよい。続いて、情報処理装置１０は、例えば図７と同様に図１３に示すように、試料Ｇのテキストデータ ＴＸ３を配列方向（アミノ酸の配列方向）にＮ行で、その配列方向に直交する非配列方向にＭ列の４文字の長さの部分テキストデータに分割する。なお、Ｎ，Ｍはそれぞれ２以上の任意の整数であり、一例としてＮ＝１６，Ｍ＝１３に設定できる。本例ではテキストデータＴＸ３に例えば１２文字分のダミーデータ（本例では０であるが、それ以外に例えば文字Ａなども使用できる）を付加して得られる８３２（＝４・１６・１３）バイトのテキストデータ（これをＴＸ３’と呼ぶ）を作成し、テキストデータＴＸ３’をＮ＝８，Ｍ＝２６で分割する。本例では、ヌクレオチドの配列を扱う場合と異なり、その４文字分の部分テキストデータをそのまま３２ビットの変換データとして扱う。なお、この際に表２に示すように、各アミノ酸を６ビットのデータ

で表してもよいが、データ量は３／４程度になるだけであるため、本例では部分テキストデータをそのまま変換データ（数値データ）として扱う。

【０１３９】

それに続いて、情報処理装置１０は、その８行で２６列の変換データに対して、図８の例と同様に、各列の変換データの配列方向に対する法2^{32}（ｍｏｄ2^{32}）のもとでの和、即ち配列方向のシンドロームを計算する。更に、各行の変換データの内で奇数番目の変換データの非配列方向に対する法2^{32}のもとでの和、及び偶数番目の変換データの非配列方向に対する法2^{32}のもとでの和、即ち非配列方向の２組のシンドロームを計算する。この例においては、シンドロームそれぞれ３２ビット（４バイト）であるため、全部のシンドロームのデータ量は、１６８（＝４・４２）バイトとなる。従って、全部のシンドロームのデータ量は、全体の元のテキストデータＴＸ３（８２０バイト）に対してほぼ１／４～１／５に減少している。

【０１４０】

次に、情報処理装置１０は、試料Ｇの名前の情報、配列の数ＮＡ３、テキストデータＴＸ３、要約値ＡＢ３，ＡＢＲ３、及びシンドロームを磁気ディスク装置１７のワーキングファイル２０に記録する。この際に、ワーキングファイル２０を複数のファイルとしてもよい。その後、情報処理装置１０は、試料Ｇの名前の情報、配列の数ＮＡ３、配列ＳＴ３，ＳＢ３、要約値ＡＢ３、逆方向の配列ＳＴＲ３，ＳＢＲ３、及び逆方向の要約値ＡＢＲ３を磁気ディスク装置１７のコンテンツファイル２１に記録する。更に、情報処理装置１０は、コンテンツファイル２１中の情報を通信ネットワーク１を介してコンテンツのプロバイダ３に送信する。これによって、コンテンツファイル２１中の情報はプロバイダ３のサーバ内の閲覧可能なコンテンツファイル３１に記録されて、第３者がインターネットを介して自由に閲覧できるようになる。この結果、第３者は、公開されている試料Ｇの配列の数ＮＡ３、及び要約値ＡＢ３（又は必要に応じてＡＢＲ３）を自分の保有するアミノ酸の配列の配列数、及び要約値と比較することによって、その試料Ｇが自分にとって新規かどうかを判定できる。また、ユーザは、その試料Ｇの配列情報を複数の供給者から誤って重複して購入することを回避することができ

る。

【０１４１】

その後、コンピュータシステム２Ａの所有者（アミノ酸情報の供給者）は、ユーザから購入要求が来るのを待つ状態となる。そして、ユーザから試料Ｇに対する簡易データの要求があったときには、情報処理装置１０は、磁気ディスク装置１７のワーキングファイル２０の中の試料Ｇのシンドロームの情報を例えば電子メールの添付ファイルとしてそのユーザに送信する。シンドロームの情報を購入したユーザは、試料Ｇと同じ種類の自分で解読した試料のアミノ酸の配列のシンドロームと、その購入したシンドロームとを比較することによって、相違する部分の検出及び復元を或る程度行うことができる。

【０１４２】

一方、ユーザから完全データの要求があったときには、情報処理装置１０は、ワーキングファイル２０中のテキストデータＴＸ３をＺＩＰファイル等の形式で圧縮し、この圧縮されたデータを例えば電子メールの添付ファイルとしてそのユーザに送信する。この際に必要に応じて、ハッシュ関数による要約値ＡＢ３を同時に送信してもよい。本例によれば、簡易データ（シンドローム）はデータ量が少ないために短時間で送信することができる。

【０１４３】

更に、そのアミノ酸の配列情報の供給者は、ワーキングファイル２０に記録した情報、即ち試料Ｇの名前の情報、配列の数ＮＡ３、テキストデータＴＸ３、要約値ＡＢ３，ＡＢＲ３、及びシンドロームをＣＤ－Ｒ／ＲＷドライブ１５を介してＣＤ－Ｒ１６に記録してもよい。このＣＤ－Ｒ１６から、更に多数のＣＤ－ＲＯＭを作製してもよく、これらの記録媒体が郵送等によってユーザに販売される。

【０１４４】

次に、本例において、アミノ酸の配列中から所望の連続する部分的な配列を選択する方法の一例につき説明する。そのため、図１３の試料Ｇの配列が、図１の表示装置１２の表示画面中に表示されているものとして、その表示画面の右端のエッジ部を図１３のエッジ部５１とする。

図１３において、試料Ｇのアミノ酸の配列はエッジ部５１の左側の表示領域に表示されており、その表示領域には図１のマウス２０４によって制御されるカーソル５２も表示されている。この場合、図１３の第１６列の第２行～第７行の矩形枠で囲まれた領域５４内の配列を選択するものとすると、本例では先ず領域５４の右端部の文字”Ａ”の上にカーソル５２を移動して、図１のマウス２０４の左スイッチ２０４ａを操作する。その後、カーソル５２がエッジ部５１から更に右方向の位置５３まで仮想的に移動するように、マウス２０４を右方向に移動する。

【０１４５】

本例では、そのようにカーソル５２が一方のエッジ部に達した後にも、更にカーソル５２が表示領域の外側に移動するようにマウス２０４を移動すると、そのカーソル５２は、そのエッジ部に対向する他方のエッジ部からその表示領域内に現れるというスクリーン・ラッピング動作が行われる。この結果、カーソル５２は、図１３の表示領域の不図示の左側のエッジ部の右側に移動して、領域５４の左端部の文字”Ｋ”上に移動して、領域５４の配列が選択される。この状態で一例としてマウス２０４の右スイッチ２０４ｂを操作することによって、領域５４の配列のコピー等を行うことができる。

【０１４６】

次に、図１４は、図１３の配列の第１５列～第１７列の配列を示し、この図１４において、第１６列の第８行の領域５６Ａ、及びこれに続く第１７列の第１行の領域５６Ｂの配列を選択するものとする。このとき、先ず領域５６Ａの左端部の文字”Ｌ”の上にカーソル５２を移動して、図１のマウス２０４の左スイッチ２０４ａを操作する。その後、カーソル５２がエッジ部５１から更に右下方向の位置５５まで仮想的に移動するように、マウス２０４を右下方向に移動する。この結果、スクリーン・ラッピング動作によって、本例のカーソル５２は、図１４の表示領域の不図示の左側のエッジ部の右側に移動して、領域５６Ｂの右端部の文字”Ｌ”上に移動して、領域５６Ａ，５６Ｂの配列が選択される。この状態で一例としてマウス２０４の右スイッチ２０４ｂを操作することによって、領域５６Ａ，５６Ｂの配列のコピー等を行うことができる。

【０１４７】

このように本例によれば、カーソルのスクリーン・ラッピング動作によって、マウス２０４の移動量を少なくしてアミノ酸の配列中の一連の広い領域、及び左右に離れた端部の連続する領域の配列を容易に選択することができる。同様に、ヌクレオチドの配列中から所定の部分的な領域を選択する場合にも、カーソルのスクリーン・ラッピング動作を行うことによって、選択動作を容易にかつ高速に行うことができる。

【０１４８】

次に、カーソルのスクリーン・ラッピング動作の別の例につき図１７を参照して説明する。この例では、ユーザが図１のマウス２０４を用いて所望のアプリケーション・プログラムを起動する際の動作につき説明する。ここでは、図１の表示装置１２の表示領域を図１７の表示領域２０１ａとして、表示領域２０１ａの長辺方向をｘ方向、短辺方向をｙ方向とする。また、カーソルの座標を指定できる範囲を有効座標領域２０１ｂとする。この場合、カーソルの座標を表示領域２０１ａの外側で、かつ有効座標領域２０１ｂの内側に設定すると、カーソルは表示領域２０１ａのエッジ部に表示される。

【０１４９】

図１７（ａ）は、表示領域２０１ａに表示されるプログラムリストの一例を示し、この図１７（ａ）の表示領域２０１ａには、メニューリスト２２１から選択されたプログラムの第１のグループリスト２２２（第１列）、及び第２のグループリスト２２３（第２列）がｘ方向に２列に分けて表示されている。この表示は、メニューリスト２２１上で「プログラム」の表示（反転している）上をカーソル２２０が通過することによって生成される。本例では、第２のグループリスト２２３中のグループＧ１６中の或るアプリケーション・プログラムを実行したいものとして、カーソル２２０を「グループＧ１６」の表示（反転している）上に移動させる。本例の表示領域２０１ａでは、第２のグループリスト２２３（第２列）の右側（＋ｘ方向）にはサブ情報を表示する余地が無いため、グループＧ１６のアプリケーションリスト２２４は、グループリスト２２２（第１列）の左側（－ｘ方向）に表示される。ここで、実行したいアプリケーション・プログラム

がアプリケーションＡ３であるとすると、どのようにカーソル２２０をアプリケーションリスト２２４上に移動するかが問題となる。

【０１５０】

即ち、単にカーソル２２０をグループＧ１６上から左側のグループリスト２２２上に移動すると、例えばグループＧ２のアプリケーションリストが表示され、グループＧ１６のアプリケーションリスト２２４の表示が消えてしまう。そこで、本例では、カーソル２２０をグループＧ１６上から右方向（＋ｘ方向）に移動させる。そして、カーソル２２０の座標をＰ（ｍ，ｎ）とすると、更にカーソル２２０の座標が、表示領域２０１ａを囲む有効座標領域２０１ｂの＋ｘ方向の外側の座標Ｐ（ｍ１，ｎ１）となるように、図１のマウス２０４を右方向に移動する。

【０１５１】

この結果、カーソル２２０は、グループＧ１６の表示の右側の位置から、図１７（ｂ）に示すように座標Ｐ（０，ｎ１）の位置の近傍、即ちアプリケーションリスト２２４上に移動する。この後、マウス２０４を僅かに下方向に移動して、カーソル２２０をアプリケーションＡ３の表示上に移動させた状態で、図１の左スイッチ２０４ａをクリックすることによって、極めて短時間に、かつ容易にアプリケーションＡ３のプログラムを起動することができる。

【０１５２】

次に、本例では、例えば図１７（ａ）において、カーソル２２０の計算上の座標が表示領域２０１ａの外部で、かつ有効座標領域２０１ｂの内部にある（カーソル２２０は表示領域２０１ａのエッジ部に表示されている）とき、即ちカーソル２２０が不活性である（アイドリング状態にある）ときには、図１のマウス２０４のスイッチ２０４ａ，２０４ｂに別の機能を持たせるようにしてもよい。このようにスイッチ２０４ａ，２０４ｂに別の機能を持たせるときには、カーソル２２０の形状を変形させてもよい。一例として、そのようにカーソル２２０が不活性であるときに、左スイッチ２０４ａを操作しながらマウス２０４をドラッグしたときには、表示領域２０１ａの大きさを所定範囲で伸縮できるようにしてもよい。更に、上記の範囲のみならず、例えばカーソル２２０が表示領域２０１ａ

の輪郭（エッジ部）に対して内側に隣接する幅Ｌ１の枠状の領域にあるときにも、スイッチ２０４ａ，２０４ｂに対して別の機能を持たせてもよい。

【０１５３】

以上をまとめると、本例による情報選択方法は、複数の情報（２２１～２２３）が表示された表示領域（２０１ａ）より、その複数の情報の何れか、又はその複数の情報の何れかに関連する情報を選択する情報選択方法において、移動量及び移動方向の少なくとも一方の情報を含む制御情報を生成し、この生成された制御情報に基づいてその表示領域内にその複数の情報に重畳させて移動自在にカーソル（２２０）（ポインタ）を表示し、このカーソルとその複数の情報の表示との位置関係に基づいて、その複数の情報の何れか、又はその複数の情報の何れかに関連する情報を選択できるようにしておき、そのカーソルをその表示領域の周縁部の第１の端部に移動させた状態で、更にそのカーソルをその表示領域の外側に移動させるようにその制御情報を与えたときに、そのカーソルをその表示領域のその周縁部のその第１の端部とは異なる第２の端部を起点としてその表示領域内で移動させるものである。

【０１５４】

即ち、本例のカーソル（２２０）は、ポインティングデバイスの制御情報に応じて、スクリーン・ラッピング方式で表示領域（２０１ａ）中を周期的に移動する。

この結果、ＧＵＩ(Graphical User Interface)方式でコンピュータ等の各種装置を操作する際に、登録してあるアプリケーション・プログラムの個数が多い場合でも、高速にカーソルを所望のアプリケーション・プログラムの位置に移動させて、そのプログラムを起動することができる。

【０１５５】

本例において、その表示領域（２０１ａ）が所定の軸に関して実質的に軸対称の領域（矩形領域、又は楕円形の領域等）である場合、その第２の端部は、その表示領域内でその所定の軸に関してその第１の端部と実質的に軸対称の位置に設定されると共に、そのカーソル（２２０）のその第２の端部からの移動方向は、その制御情報によってその第１の端部から更にそのカーソルを移動させようとし

た方向であることが望ましい。これによって、カーソル（２２０）の周期的な動きが容易に予測できるため、ユーザが特に習熟することなく、直ぐにその周期的なカーソル（２２０）の動きを活用できる。

【０１５６】

また、そのカーソル（２２０）をその表示領域（２０１ａ）のその周縁部の第１の幅の制限領域に移動させた状態、及びそのカーソルをその制限領域から更に第２の幅（Ｌ）だけ外側（２０１ｂ）に移動させるようにその制御情報を与えた状態では、そのカーソルに対してその情報の選択以外の別の機能を与えることが望ましい。通常は、その表示領域（２０１ａ）の周縁部にはアプリケーション・プログラムのアイコン等は表示されていない。そこで、そのカーソル（２２０）がその表示領域（２０１ａ）の周縁部に有る状態では、アプリケーション・プログラムの選択以外の機能、例えばその表示領域の伸縮機能等を持たせても、アプリケーション・プログラムの選択には実質的に影響が無いと共に、カーソル（２２０）（ポインティングデバイス）の用途が広がる利点がある。

【０１５７】

また、本例では図１７に示すように、その表示領域（２０１ａ）内にその複数の情報、及びこれらの情報に関連する情報が複数列（２２２，２２３）に表示されているときに、そのカーソル（２２０）がその複数列の一方の端部の列（２２３）の所定の情報の表示を通過しているときに、その表示領域内のその複数列の他方の端部の列（２２２）の外側にその所定の情報に関連する複数のサブ情報（２２４）を表示し、更にそのカーソルをその複数列の一方の端部の列（２２３）からその表示領域の外側に移動させるようにその制御情報を与えたときに、そのカーソル（２２０）をその表示領域のその複数列の他方の端部の列（２２２）に近い端部Ｐ（０，ｎ１）を起点として、その複数のサブ情報（２２４）の表示上に移動させて、その複数のサブ情報の何れかを選択可能としている。

【０１５８】

即ち、図１７に示すように、その表示領域（２０１ａ）内に表示すべきアプリケーション・プログラムの個数が多い場合には、例えばその右側の端部の列（２２３）のサブ情報（２２４）が左側の端部の列（２２２）の外側に表示される。

このときに、本例の周期的な移動を行うカーソル（２２０）を適用すると、そのサブ情報（２２４）中のアプリケーション・プログラムを選択するためには、ポインティングデバイスによってそのカーソル（２２０）をその列（２２３）から更に右方向に移動させるようにすればよい。これによって、アプリケーション・プログラムの個数が多く、プログラムリストが複数列になるような場合でも、ＧＵＩ方式で容易にカーソルを所望のアプリケーション・プログラムの位置に移動できる。

【０１５９】

ここで、ＤＮＡ又はＲＮＡのヌクレオチドの配列（塩基の配列）に対応するテキストデータ（又はこれを表１等に基づいて変換した数値データ）の要約値を算出するためのハッシュ関数について更に説明する。例えばハッシュ関数の演算対象を、人間のＤＮＡのヌクレオチドの配列とすると、そのテキストデータ又は数値データのファイル（以下、「原ファイル」と言う）は１００Ｍバイト程度にも達する膨大なファイルである。そこで、本発明で使用するハッシュ関数（ハッシュ演算アルゴリズム）は、演算対象の原ファイルを分割した後の複数の分割ファイルを順次処理することによって、全体の要約値を算出する機能を持つことが望ましい。

【０１６０】

また、ハッシュ関数は、一例として所定ビット数ｍ１（ｍ１は例えば３２，６４等）のデータを１ワードとして、所定ワード数ｍ２（ｍ２は例えば１６，３２，６４等）単位で、原ファイルの要約値を算出していく。この際に、データの処理単位は、ｍ１・ｍ２ビットとなる。例えばｍ１＝３２，ｍ２＝１６では、処理単位は５１２ビットとなる。そこで、その原ファイルを複数の分割ファイルに分割する際には、最初はｍ１・ｍ２ビットの整数倍（例えば１００００倍程度）を単位として分割していき、端数として残ったデータに所定データ（長さを表すデータ、区切りデータ等）を付加してｍ１・ｍ２ビットの整数倍のファイルとすることで、要約値の演算を効率的に実行することができる。

【０１６１】

更に、暗号理論で使用されるハッシュ関数は、テキストデータ中のスペースコ

ード及び改行コード等も全て演算処理対象としているが、ヌクレオチド及びアミノ酸の配列情報については見やすくするために、例えば配列番号１～３で示すように、途中にスペースコード、順序を示す数字コード、及び改行コードを挿入する場合がある。そこで、ヌクレオチドの配列情報（アミノ酸の配列情報も同様）を演算処理対象とするハッシュ関数においては、テキストデータ中の所定コードとしてのスペースコード、数字コード、及び改行コードを無視する機能を付加することが望ましい。また、隣接する文字を”－”（ハイフン）で分けることも考えられるが、この場合には、更に”－”記号も無視する必要がある。

【０１６２】

更に、原ファイルを複数の分割ファイルに分割する際には、複数の分割ファイルの順序等を示すデータ（以下、「コメントデータ」と言う）を各分割ファイルに付加することが望ましいことがある。このように分割ファイル、又は１つの原ファイルにコメントデータを付加する場合にも、コメントデータはハッシュ関数で無視する必要がある。そのため、例えばコメントデータは所定の開始記号（例えば　／＊　）及び終了記号（例えば　＊／）の間に記録し、ハッシュ関数で処理する際に開始記号から終了記号までのデータは無視するようにすればよい。

【０１６３】

また、上記の実施の形態では、例えば生物のＤＮＡのヌクレオチドの配列（又はタンパク質のアミノ酸の配列）内の先頭の一部、及び末尾の一部の配列、並びにその配列のテキストデータの要約値をインターネット上で公開することがある。この場合には、その公開されている一部の配列と、その要約値とからそのテキストデータの内容が推定される可能性もある。これを回避するために、そのテキストデータをハッシュ関数で処理する際に、その公開されている配列を除いた部分についてのみ、そのハッシュ関数を施して要約値を求めるようにしてもよい。

【０１６４】

次に、例えば核酸や遺伝子のヌクレオチドの配列が見易いように順序を示す数字、スペース、及び改行を含んでテキストデータとして記録されたファイル（ファイルＦＤ１とする）の要約値(message digest)を計算するための方法の一例につき図１５を参照して説明する。なお、以下の要約値の計算は、例えば図１の情

報処理装置１０において実行される。

【０１６５】

図１５において、先ずステップ１５１では、ファイルＦＤ１中のテキストデータから数字コード、スペースコード、及び改行コードを取り除いたテキストデータをファイルＦＤ２に記録する。その次のステップ１５２では、例えばＭＤ５ハッシュ関数を用いてファイルＦＤ２中のテキストデータの１２８ビットの要約値を算出する。この方法は処理は単純であるが、ファイルＦＤ１が例えば１００Ｍバイト程度であるとすると、ファイルＦＤ２もほぼ１００Ｍバイト程度になるため、記憶装置の容量を大きくする必要がある。

【０１６６】

ＭＤ５ハッシュ関数のアルゴリズムについては、上記のウェブサイト２で詳細に開示されているが、ここでそのアルゴリズムについて簡単に説明する。

先ず、ここでは１ワード"word"とは、３２ビットの量であり、１バイト"byte"とは、８ビットの量である。そして、一列のビットは、自然に一列のバイトと解釈することができ、ここではそれぞれ８ビットのデータの集まりを、ＭＳＢ(most significant bit)、即ち上位ビットが最初に表示される１バイトのデータとして解釈することができる。同様に、一列のバイトは、一列の３２ビットのワードと解釈することができ、ここではそれぞれ４バイトのデータの集まりを、ＬＳＢ(least significant byte)、即ち下位バイトが最初に表示される１ワードのデータとして解釈することができる。

【０１６７】

また、次のように演算等を定義する。即ち、" x_i " は x に下付き文字 i を付加した表現を意味し、その下付き文字が一つの式であるときには、例えば" x_[i+1]"のようにその式を括弧で囲むものとする。同様に、上付き文字（べき乗）としては ^ を用いる。従って、"x^i" はx のi 乗を意味する。

また、記号"+" は、ワードの加算、即ち法 2^32 の加算を意味する。そして、" X <<< s " は、X をs ビットだけ左側に循環的にシフトして（回転して）得られる３２ビットの値を意味する。また、 not(X) は、X のビット毎の補数(complement)を意味し、" X v Y"は、X とY とにビット毎のＯＲ演算を施して得られる

値を意味し、” X xor Y”はX とY とにビット毎のＸＯＲ（排他的論理和）演算を施して得られる値を意味し、”XY”はX とY とにビット毎のＡＮＤ演算を施して得られる値を意味する。

【０１６８】

次に、上記のファイルＦＤ２に記録されているテキストデータ（ファイルＦＤ１から数字コード、スペースコード、及び改行コードを取り除いたテキストデータ）を、要約値を求めるべきｂビットのメッセージであるとする。その値ｂは、任意の非負整数であり、ｂは０であってもよい。その値ｂは８の倍数である必要はなく、更に任意に大きい値であってもよい。そのｂビットのメッセージの一連のビットは次のように表すことができる。

【０１６９】

m_0 m_1 ... m_[b-1]

そのメッセージの要約値は、次の５つのステップＡ～Ｅの処理で計算することができる。

［ステップＡ］（追加ビットの付加）

そのメッセージには、そのメッセージをビット列で表現したときの長さが法５１２のもとで４４８に合同となるように追加ビットが付加（拡張）される。即ち、そのメッセージは、その長さが５１２の倍数のビットよりも６４ビットだけ少ない長さになるように拡張される。追加ビットの付加は、たとえそのメッセージの長さが既に法５１２のもとで４４８に合同である場合でも常に実行される。

【０１７０】

追加ビットの付加は、単一のビット”1” を付加した後に、ビットの長さが法５１２のもとで４４８に合同となるようにビット”0” を付加することによって実行される。全ての場合に、少なくとも１ビット、そして最大で５１２ビットが付加される。

［ステップＢ］（長さ情報の付加）

そのメッセージのビット数であるｂ（ステップＡにおける追加ビットの付加が行われる前の長さ）の６４ビットの表現が、ステップＡで得られたメッセージに付加される。実際には起こりそうもないが、仮にｂが６４ビットよりも大きいと

きには、ｂの表現の下位の６４ビット分だけが付加される。これらのビットは、２つの３２ビットのワードとして、上述の内容に対応して下位のワードが最初になるように付加される。

【０１７１】

このようにして得られたメッセージのビット表現の長さは、正確に５１２の倍数、即ち５１２ビットの倍数となる。言い換えると、このようにして得られたメッセージの長さは、正確に１６個の（３２ビットの）ワードの倍数となる。そこで、このようにして得られたメッセージの各ワードを M[0 ... N-1] とする。ここで、Ｎは１６の倍数である。

【０１７２】

［ステップＣ］（要約値バッファの初期化）

要約値を計算するために４ワードのバッファ（A,B,C,D)を使用する。ここで、A，B，C，D はそれぞれ３２ビットのレジスタであり、これらのレジスタは下位バイトを最初に記載する１６進表現で次の値に初期化される。

ワード A: 01 23 45 67

ワード B: 89 ab cd ef

ワード C: fe dc ba 98

ワード D: 76 54 32 10

［ステップＤ］１６ワードブロック毎のメッセージの処理

ここでは、それぞれ入力として３個の３２ビットのワードを受け取って出力として１個の３２ビットのワードを生成する４個の補助的な関数を次のように定義する。

【０１７３】

F(X,Y,Z) = XY v not(X) Z

G(X,Y,Z) = XZ v Y not(Z)

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X v not(Z))

各ビット位置で、関数Ｆは、Ｘが真ならばＹで、そうでなければＺという条件式として作用する。関数Ｆは、v の代わりに＋を使って定義することもできた。

なぜなら、 XY と not(X)Z とは、同じビット位置で共に１となることが決してないからである。

【０１７４】

関数Ｇ，Ｈ，Ｉは、X,Y,Z のビットからビット毎に並行に出力を生成する点で関数Ｆと同様である。

また、関数Ｈは、その入力に対してビット毎のＸＯＲ演算、又はパリティ演算を施す関数である。

更にこのステップＤでは、正弦関数から導かれる６４個の要素を持つテーブル T[1 ... 64] を用いる。即ち、そのテーブルのｉ番目の要素を T[i] として、ｉの単位をラジアンとすると、次のようになる。

【０１７５】

T[i]＝ {4294967296 × abs(sin(i))} の整数部

なお、abs(sin(i)) は sin(i) の絶対値である。これらの関数及びテーブルを用いて以下の演算を行う。

各１６ワードのブロックを処理するために、変数ｉについて０から（Ｎ／１６－１）まで以下の［ｉに関するループの始まり］から［ｉに関するループの終わり］までの処理を繰り返して行う。

【０１７６】

［ｉに関するループの始まり］

先ず変数ｊについて０から１５まで、１ワードのメッセージ M[i*16+j] を X[j] にコピーする。

続いて、バッファA,B,C,D の値をそれぞれ次のようにバッファAA,BB,CC,DD にコピーする。

【０１７７】

AA = A, BB = B, CC = C, DD = D

［ラウンド１］

ここで、[abcd k s i] は次の処理を行うものと定義する。

a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s)

そして、次の１６回の処理を行う。

【０１７８】

[ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]

[ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]

[ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]

[ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]

[ラウンド２]

ここで、[abcd k s i] は次の処理を行うものと定義する。

【０１７９】

a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s)

そして、次の１６回の処理を行う。

[ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]

[ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]

[ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]

[ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]

[ラウンド３]

ここで、[abcd k s t] は次の処理を行うものと定義する。

【０１８０】

a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s)

そして、次の１６回の処理を行う。

[ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]

[ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]

[ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]

[ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]

[ラウンド４]

ここで、[abcd k s t] は次の処理を行うものと定義する。

【０１８１】

a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s)

そして、次の１６回の処理を行う。

[ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]

[ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]

[ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]

[ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]

次に、バッファA,B,C,D の値にそれぞれ次のようにバッファAA,BB,CC,DD の値（このブロックの処理が始まる前のバッファA,B,C,D の値）を加算する。

【０１８２】

A = A + AA, B = B + BB, C = C + CC, D = D + DD

［ｉに関するループの終わり］

［ステップＥ］出力

出力として計算された要約値はバッファA，B，C，D　の値そのものである。即ち、バッファA の下位バイトから始まって、バッファD の上位バイトで終わる値がその要約値である。なお、要約値が３２ビット又は６４ビットでよいような場合には、それぞれ例えばバッファA 、又はバッファA,B の値のみを要約値として用いてもよい。

【０１８３】

また、ＭＤ５ハッシュ関数は、元のデータの推定が困難となるように複雑な処理を行っているが、ヌクレオチドやアミノ酸の配列データの要約値を計算する場合には元のデータが或る程度推定されても特に不都合がないことがある。この場合には、メッセージに対応する一連の所定ビット数ｓ（ｓはＭＤ５ハッシュ関数では５１２）のブロックＢｉ（ｉ＝１～Ｉ）毎の演算を、順次次のような簡単な演算で行うことも考えられる。

【０１８４】

$M_1 = (a \cdot B1 + b) \bmod 2^s$

$M_i = (M_{i-1} \cdot Bi + b) \bmod 2^s$ （ｉ＝２～Ｉ）

この場合、ａ，ｂは０以外のｓビットの数であり、M_I が最終的な要約値となる。

次に、上記のファイルＦＤ１の要約値を計算するための別の方法につき図１６を参照して説明する。ここではＭＤ５ハッシュ関数を用いて要約値を計算するものとする。以下の計算も一例として図１の情報処理装置１０で実行される。

【０１８５】

先ず図１６のステップ１６１において、ヌクレオチド自体を表すコードの個数を表す変数ＮＸ，ＮＹの値をそれぞれ０に設定し、要約値を表す３２ビットずつのバッファＡ，Ｂ，Ｃ，Ｄの値を所定の初期値（上記のステップＣで設定した値）に設定し、要約値の計算対象のテキストデータを空にする。

次のステップ１６２において、ファイルＦＤ１中のテキストデータの先頭から１文字分（ここでは１バイト）の文字コード（ここでは全ての種類のコードを含む意味である）を読み取り、それに続くステップ１６３において、読み取った文字コードが数字コード、スペースコード、又は改行コードかをチェックする。そして、読み取った文字コードが数字、スペース、改行の何れのコードでもない、即ち本例ではＡ～Ｚ，ａ～ｚの何れかのコードであるときには、ステップ１６４に移行して、変数ＮＸの値に１を加算すると共に、読み取った文字コードを要約値の計算対象のテキストデータに加える。続いてステップ１６５において、変数ＮＸ（読み取られた有効な文字コードの個数）が、要約値の計算単位である文字数N_A に達したかどうかを調べる。本例では、N_A ＝５１２／８＝６４である。

【０１８６】

ＮＸ＝N_A であるときには、ステップ１６６に移行して、変数ＮＸを０に戻すと共に、変数ＮＹ（N_A 個の文字単位のブロック数）に１を加算した後、ステップ１６７に移行して、計算対象のテキストデータ（N_A 個の文字コードを含んでいる）の要約値（Ａ，Ｂ，Ｃ，Ｄ）を計算する。これは、上記のステップＤを１回実行することを意味する。その後、計算対象のテキストデータを空にしてから、動作はステップ１６８に移行して、ファイルＦＤ１中に読み取り対象となる文字コードがまた有るかどうかがチェックされる。また、ステップ１６３で読み取られた文字コードが数字、スペース、改行の何れかのコードであるときには、ステップ１６９で読み取った文字コードを無視した後、ステップ１６８に移行する。更に、ステップ１６５で変数ＮＸがN_A に達していないときにも、動作はステップ１６８に移行する。

【０１８７】

そして、ステップ１６８において、読み取り対象となる文字コードがまだ有る

ときには、動作はステップ１６２に戻り、ファイルＦＤ１中のテキストデータから次の１文字分の文字コードが読み取られて、以下ステップ１６３～１６８の動作が繰り返される。一方、ステップ１６８において、読み取り対象となる文字コードが無くなったときには、動作はステップ１７０に移行して、要約値（Ａ，Ｂ，Ｃ，Ｄ）が計算される。この際に、変数ＮＸ，ＮＹの値より読み取られた有効な文字コードの全個数が分かるため、上記のステップＡ、ステップＢ、ステップＤ、ステップＥが実行される。得られた要約値（Ａ，Ｂ，Ｃ，Ｄ）が最終的な要約値となる。

【０１８８】

具体的にＭＤ５ハッシュ関数を用いて図１５、及び図１６の計算方法で、配列番号１、２のヌクレオチドの配列を示すテキストデータ、及び配列番号３のアミノ酸の配列を示すテキストデータの要約値を計算した結果は、１６進数表示で以下のようになる。これらの要約値は、配列の改行方法等を変えても変化しない一定の値である。

【０１８９】

ＭＤ５の要約値（配列番号１）＝hex(1c0a0b1d72e256bb10556a2fb52d28ae)

ＭＤ５の要約値（配列番号２）＝hex(ec8c3c9af5630f61f3d0cd2bd13b0f0d)

ＭＤ５の要約値（配列番号３）＝hex(164f14406ac21158e20ba72666a033ab)

この要約値の計算方法によれば、ファイルＦＤ１から逐次関係の無い文字コードを取り除きながら要約値を計算しているため、記憶装置の記憶容量を殆ど増加する必要がないという利点がある。従って、ファイルＦＤ１の情報量が大きくなる程、この計算方法は有利になる。また、上記のステップ１５１、１６３では、所定コードとしての数字コード、スペースコード、改行コードを取り除いているが、それ以外にコメント文などを取り除くようにしてもよい。

【０１９０】

なお、本発明は上述の実施の形態に限定されず、本発明の要旨を逸脱しない範囲で種々の構成を取り得ることは勿論である。

【０１９１】

【発明の効果】

本発明によれば、核酸や遺伝子中のヌクレオチドの配列情報、又はタンパク質やペプチド中のアミノ酸の配列情報を、それらの配列が所定の長さを超えたときに、それらの配列を示すテキストデータよりも少ないデータ量で記録することができる。従って、それらの配列情報を通信回線を介して短時間に送信することが可能となる。

【０１９２】

また、それらの配列を示すテキストデータ、又はこれに対応する数値データの数学的な要約値を用いた場合には、膨大な長さの２つのヌクレオチドの配列同士、又は２つのアミノ酸の配列同士の同一性を少ないデータ量で高精度に確認することができる。また、同一の複数の配列情報を誤って購入することも防止できる。

【０１９３】

また、シンドローム情報を用いた場合には、２つのヌクレオチドの配列（又は２つのアミノ酸の配列）の間の相違する部分を少ないデータ量で容易に検出できると共に、必要に応じてその相違する部分の情報を復元することができる。従って、例えばＳＮＰ（一塩基変位多型：Single Nucleotide Polymorphism）を少ないデータ量で容易に発見することができる。

【０１９４】

また、本発明によれば、ヌクレオチドの配列情報、又はアミノ酸の配列情報を少ないデータ量でユーザに供給できるビジネスモデルを提供することができる。この場合に、更に数学的な要約値、又はシンドローム情報を用いることによって、ユーザが提供された配列情報と情報供給者が保持している配列情報との同一性の確認、又は相違する部分の検出や復元を容易に行うことができる。

【０１９５】

また、本発明の要約値の計算方法によれば、例えばヌクレオチドの配列を見易くするための数値コードやスペースコードなど、又はその配列の内容を説明するためのコメント文などの所定コードを無視して、必要な情報のみの要約値を算出できるため、その所定コードの内容が変化しても、常に同一の要約値を算出できる利点がある。従って、その要約値の計算方法は、特にヌクレオチドやアミノ酸

の配列情報の要約値を算出する場合に有効である。

【図面の簡単な説明】

【図１】　本発明の実施の形態の一例で使用されるコンピュータシステムを示す概略構成図である。

【図２】　その実施の形態の一例で処理対象とするＤＮＡ、及びそのヌクレオチドの配列のバイナリーデータによる表現の例を示す図である。

【図３】　その実施の形態の一例におけるＤＮＡ情報の供給者の動作の一部を示すフローチャートである。

【図４】　図３の動作に続くＤＮＡ情報の供給者の動作を示すフローチャートである。

【図５】　その実施の形態の一例におけるＤＮＡ情報のユーザの動作の一部を示すフローチャートである。

【図６】　図５の動作に続くＤＮＡ情報のユーザの動作を示すフローチャートである。

【図７】　標準試料Ｅ（ＤＮＡ）のヌクレオチド（２０４８個）の配列を表すテキストデータを４行で３２列の部分テキストデータＴ（ｉ，ｊ）に分割した状態を示す図である。

【図８】　標準試料Ｅの変換データＡ（ｉ，ｊ）、及びこれらから算出されるシンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）を示す図である。

【図９】　標準試料ＥのシンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）を示す図である。

【図１０】　試料Ｆ（ＤＮＡ）のヌクレオチド（２０４８個）の配列を表すテキストデータを４行で３２列の部分テキストデータＴＦ（ｉ，ｊ）に分割した状態を示す図である。

【図１１】　試料Ｆの変換データＡＦ（ｉ，ｊ）、及びこれらから算出されるシンドロームＣＦ（ｊ），Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）を示す図である。

【図１２】　試料ＦのシンドロームＣＦ（ｊ），Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）、及び復元された変換データを示す図である。

【図１３】　試料Ｇ（タンパク質）のアミノ酸（８２０個）の配列を表すテ

キストデータを８行で２６列の部分テキストデータに分割した状態を示す図である。

【図１４】　図１３中の一部のデータを示す図である。

【図１５】　本発明の実施の形態の第１の要約値計算シーケンスを示すフローチャートである。

【図１６】　本発明の実施の形態の第２の要約値計算シーケンスを示すフローチャートである。

【図１７】　本発明の実施の形態の表示画面内でのカーソルの移動方法の一例を示す図である。

【符号の説明】

１…通信ネットワーク、２Ａ，２Ｂ…コンピュータシステム、３…コンテンツのプロバイダ、４…ＤＮＡのシーケンサー、１０…情報処理装置、１５…ＣＤ－Ｒ／ＲＷドライブ、１６…ＣＤ－Ｒ、１７…磁気ディスク装置、１９…マスターファイル、２０…ワーキングファイル、２１…コンテンツファイル、３１…コンテンツファイル

【配列表】

SEQUENCE LISTING

<110> Omori, Satoshi

<120> Method and apparatus for recording information of nucleotide sequence and amino acid sequence

<130> 2001A09

<140>

<141>

<150> JP 2000-117343

<151> 2000-04-19

<150> JP 2000-149122

<151> 2000-05-19

<160> 3

<170> PatentIn Ver. 2.0

<210> 1

<211> 2048

<212> DNA

<213> Escherichia coli

<400> 1

agcttttcat tctgactgca acgggcaata tgtctctgtg tggattaaaa aaagagtgtc 60

tgatagcagc ttctgaactg gttacctgcc gtgagtaaat taaaatttta ttgacttagg 120

tcactaaata ctttaaccaa tataggcata gcgcacagac agataaaaat tacagagtac 180

acaacatcca tgaaacgcat tagcaccacc attaccacca ccatcaccat taccacaggt 240

aacggtgcgg gctgacgcgt acaggaaaca cagaaaaaag cccgcacctg acagtgcggg 300

ctttttttt cgaccaaagg taacgaggta acaaccatgc gagtgttgaa gttcggcggt 360

acatcagtgg caaatgcaga acgttttctg cgtgttgccg atattctgga aagcaatgcc 420

aggcaggggc aggtggccac cgtcctctct gcccccgcca aaatcaccaa ccacctggtg 480

gcgatgattg aaaaaaccat tagcggccag gatgctttac ccaatatcag cgatgccgaa 540

cgtatttttg ccgaactttt gacgggactc gccgccgccc agccggggtt cccgctggcg 600

caattgaaaa ctttcgtcga tcaggaattt gcccaaataa aacatgtcct gcatggcatt 660

agtttgttgg ggcagtgccc ggatagcatc aacgctgcgc tgatttgccg tggcgagaaa 720

atgtcgatcg ccattatggc cggcgtatta gaagcgcgcg gtcacaacgt tactgttatc 780
gatccggtcg aaaaactgct ggcagtgggg cattacctcg aatctaccgt cgatattgct 840
gagtccaccc gccgtattgc ggcaagccgc attccggctg atcacatggt gctgatggca 900
ggtttcaccg ccggtaatga aaaaggcgaa ctggtggtgc ttggacgcaa cggttccgac 960
tactctgctg cggtgctggc tgcctgttta cgcgccgatt gttgcgagat ttggacggac 1020
gttgacgggg tctatacctg cgacccgcgt caggtgcccg atgcgaggtt gttgaagtcg 1080
atgtcctacc aggaagcgat ggagctttcc tacttcggcg ctaaagttct tcacccccgc 1140
accattaccc ccatcgccca gttccagatc ccttgcctga ttaaaaatac cggaaatcct 1200
caagcaccag gtacgctcat tggtgccagc cgtgatgaag acgaattacc ggtcaagggc 1260
atttccaatc tgaataacat ggcaatgttc agcgtttctg gtccggggat gaaagggatg 1320
gtcggcatgg cggcgcgcgt ctttgcagcg atgtcacgcg cccgtatttc cgtggtgctg 1380
attacgcaat catcttccga atacagcatc agtttctgcg ttccacaaag cgactgtgtg 1440
cgagctgaac gggcaatgca ggaagagttc tacctggaac tgaaagaagg cttactggag 1500
ccgctggcag tgacggaacg gctggccatt atctcggtgg taggtgatgg tatgcgcacc 1560
ttgcgtggga tctcggcgaa attctttgcc gcactggccc gcgccaatat caacattgtc 1620
gccattgctc agggatcttc tgaacgctca atctctgtcg tggtaaataa cgatgatgcg 1680
accactggcg tgcgcgttac tcatcagatg ctgttcaata ccgatcaggt tatcgaagtg 1740
tttgtgattg gcgtcggtgg cgttggcggt gcgctgctgg agcaactgaa gcgtcagcaa 1800
agctggctga agaataaaca tatcgactta cgtgtctgcg gtgttgccaa ctcgaaggct 1860
ctgctcacca atgtacatgg ccttaatctg gaaaactggc aggaagaact ggcgcaagcc 1920
aaagagccgt ttaatctcgg gcgcttaatt cgcctcgtga aagaatatca tctgctgaac 1980
ccggtcattg ttgactgcac ttccagccag gcagtggcgg atcaatatgc cgacttcctg 2040
cgcgaagg 2048

<210> 2
<211> 2048
<212> DNA
<213> Escherichia coli

<400> 2

agcttttcat tctgactgca acgggcaata tgtctctgtg tggattaaaa aaagagtgtc 60

tgatagcagc ttctgaactg gttacctgcc gtgagtaaat taaaatttta ttgacttagg 120

tcactaaata ctttaaccaa tataggcata gcgcacagac agataaaaat tacagagtac 180

acaacatcca tgaaacgcat tagcaccacc attaccacca ccatcaccat taccacaggt 240

aacggtgcgg gctgacgcgt acaggaaaca cagaaaaaag cccgcacctg acagtgcggg 300

ctttttttt cgaccaaagg taacgaggta acaaccatgc gagtgttgaa gttcggcggt 360

acatcagtgg caaatgcaga acgttttctg cgtgttgccg atattctgga aagcaatgcc 420

aggcaggggc aggtggccac cgtcctctct gcccccgcca aaatcaccaa ccacctggtg 480

gcgatgattg aaaaaaccat tagcggccag gatgctttac ccaatatcag cgatgccgaa 540

cgtatttttg ccgaactttt gacgggactc gccgccgccc agccggggtt cccgctggcg 600

caattgaaaa ctttcgtcga tcaggaattt gcccaaataa aacatgtcct gcatggcatt 660

agtttgttgg ggcagtgccc ggatagcatc aacgctgcgc tgatttgccg tggcgagaaa 720

atgtcgatcg ccattatggc cggcgtatta gaagcgcgcg gtcacaacgt tactgttatc 780

gatccggtcg aaaaactgct ggcagtgggg cattacctcg aatctaccgt cgatattgct 840

gagtccaccc gccgtattgc ggcaagccgc attccggctg atcacatggt gctgatggca 900

ggtttcaccg ccggtaatga aaaaggcgaa ctggtggtgc ttggacgcaa cggttccgac 960

tactctgctg cggtgctggc tgcctgttta cgcgccgatt gttgcgagat ttggacatta 1020

tggcggccaa cttatacctg cgacccgcgt caggtgcccg atgcgaggtt gttgaagtcg 1080

atgtcctacc aggaagcgat ggagctttcc tacttcggcg ctaaagttct tcacccccgc 1140

accattaccc ccatcgccca gttccagatc ccttgcctga ttaaaaatac cggaaatcct 1200

caagcaccag gtacgctcat tggtgccagc cgtgatgaag acgaattacc ggtcaagggc 1260

atttccaatc tgaataacat ggcaatgttc agcgtttctg gtccggggat gaaagggatg 1320

gtcggcatgg cggcgcgcgt ctttgcagcg atgtcacgcg cccgtatttc cgtggtgctg 1380

attacgcaat catcttccga atacagcatc agtttctgcg ttccacaaag cgactgtgtg 1440

cgagctgaac gggcaatgca ggaagagttc tacctggaac tgaaagaagg cttactggag 1500

ccgctggcag tgacggaacg gctggccatt atctcggtgg taggtgatgg tatgcgcacc 1560

ttgcgtggga tctcggcgaa attctttgcc gcactggccc gcgccaatat caacattgtc 1620

gccattgctc agggatcttc tgaacgctca atctctgtcg tggtaaataa cgatgatgcg 1680

accactggcg tgcgcgttac tcatcagatg ctgttcaata ccgatcaggt tatcgaagtg 1740
tttgtgattg gcgtcggtgg cgttggcggt gcgctgctgg agcaactgaa gcgtcagcaa 1800
agctggctga agaataaaca tatcgactta cgtgtctgcg gtgttgccaa ctcgaaggct 1860
ctgctcacca atgtacatgg ccttaatctg gaaaactggc aggaagaact ggcgcaagcc 1920
aaagagccgt ttaatctcgg gcgcttaatt cgcctcgtga aagaatatca tctgctgaac 1980
ccggtcattg ttgactgcac ttccagccag gcagtggcgg atcaatatgc cgacttcctg 2040
cgcgaagg 2048

<210> 3
<211> 820
<212> PRT
<213> Escherichia coli

<400> 3

Met Arg Val Leu Lys Phe Gly Gly Thr Ser Val Ala Asn Ala Glu Arg
1 5 10 15

Phe Leu Arg Val Ala Asp Ile Leu Glu Ser Asn Ala Arg Gln Gly Gln
20 25 30

Val Ala Thr Val Leu Ser Ala Pro Ala Lys Ile Thr Asn His Leu Val
35 40 45

Ala Met Ile Glu Lys Thr Ile Ser Gly Gln Asp Ala Leu Pro Asn Ile
50 55 60

Ser Asp Ala Glu Arg Ile Phe Ala Glu Leu Leu Thr Gly Leu Ala Ala
65 70 75 80

Ala Gln Pro Gly Phe Pro Leu Ala Gln Leu Lys Thr Phe Val Asp Gln
85 90 95

Glu Phe Ala Gln Ile Lys His Val Leu His Gly Ile Ser Leu Leu Gly
100 105 110

Gln Cys Pro Asp Ser Ile Asn Ala Ala Leu Ile Cys Arg Gly Glu Lys
115 120 125

Met Ser Ile Ala Ile Met Ala Gly Val Leu Glu Ala Arg Gly His Asn
130 135 140

Val Thr Val Ile Asp Pro Val Glu Lys Leu Leu Ala Val Gly His Tyr
145 150 155 160

Leu Glu Ser Thr Val Asp Ile Ala Glu Ser Thr Arg Arg Ile Ala Ala
165 170 175

Ser Arg Ile Pro Ala Asp His Met Val Leu Met Ala Gly Phe Thr Ala
180 185 190

Gly Asn Glu Lys Gly Glu Leu Val Val Leu Gly Arg Asn Gly Ser Asp
195 200 205

Tyr Ser Ala Ala Val Leu Ala Ala Cys Leu Arg Ala Asp Cys Cys Glu
210 215 220

Ile Trp Thr Asp Val Asp Gly Val Tyr Thr Cys Asp Pro Arg Gln Val
225 230 235 240

Pro Asp Ala Arg Leu Leu Lys Ser Met Ser Tyr Gln Glu Ala Met Glu
245 250 255

Leu Ser Tyr Phe Gly Ala Lys Val Leu His Pro Arg Thr Ile Thr Pro
260 265 270

Ile Ala Gln Phe Gln Ile Pro Cys Leu Ile Lys Asn Thr Gly Asn Pro
275 280 285

Gln Ala Pro Gly Thr Leu Ile Gly Ala Ser Arg Asp Glu Asp Glu Leu
290 295 300

Pro Val Lys Gly Ile Ser Asn Leu Asn Asn Met Ala Met Phe Ser Val
305 310 315 320

Ser Gly Pro Gly Met Lys Gly Met Val Gly Met Ala Ala Arg Val Phe
325 330 335

Ala Ala Met Ser Arg Ala Arg Ile Ser Val Val Leu Ile Thr Gln Ser
340 345 350

Ser Ser Glu Tyr Ser Ile Ser Phe Cys Val Pro Gln Ser Asp Cys Val
355 360 365

Arg Ala Glu Arg Ala Met Gln Glu Glu Phe Tyr Leu Glu Leu Lys Glu
370 375 380

Gly Leu Leu Glu Pro Leu Ala Val Thr Glu Arg Leu Ala Ile Ile Ser

385 390 395 400

Val Val Gly Asp Gly Met Arg Thr Leu Arg Gly Ile Ser Ala Lys Phe
405 410 415

Phe Ala Ala Leu Ala Arg Ala Asn Ile Asn Ile Val Ala Ile Ala Gln
420 425 430

Gly Ser Ser Glu Arg Ser Ile Ser Val Val Val Asn Asn Asp Asp Ala
435 440 445

Thr Thr Gly Val Arg Val Thr His Gln Met Leu Phe Asn Thr Asp Gln
450 455 460

Val Ile Glu Val Phe Val Ile Gly Val Gly Gly Val Gly Gly Ala Leu
465 470 475 480

Leu Glu Gln Leu Lys Arg Gln Gln Ser Trp Leu Lys Asn Lys His Ile
485 490 495

Asp Leu Arg Val Cys Gly Val Ala Asn Ser Lys Ala Leu Leu Thr Asn
500 505 510

Val His Gly Leu Asn Leu Glu Asn Trp Gln Glu Glu Leu Ala Gln Ala
515 520 525

Lys Glu Pro Phe Asn Leu Gly Arg Leu Ile Arg Leu Val Lys Glu Tyr
530 535 540

```
His Leu Leu Asn Pro Val Ile Val Asp Cys Thr Ser Ser Gln Ala Val
545                 550                 555                 560

Ala Asp Gln Tyr Ala Asp Phe Leu Arg Glu Gly Phe His Val Val Thr
                565                 570                 575

Pro Asn Lys Lys Ala Asn Thr Ser Ser Met Asp Tyr Tyr His Gln Leu
            580                 585                 590

Arg Tyr Ala Ala Glu Lys Ser Arg Arg Lys Phe Leu Tyr Asp Thr Asn
        595                 600                 605

Val Gly Ala Gly Leu Pro Val Ile Glu Asn Leu Gln Asn Leu Leu Asn
    610                 615                 620

Ala Gly Asp Glu Leu Met Lys Phe Ser Gly Ile Leu Ser Gly Ser Leu
625                 630                 635                 640

Ser Tyr Ile Phe Gly Lys Leu Asp Glu Gly Met Ser Phe Ser Glu Ala
                645                 650                 655

Thr Thr Leu Ala Arg Glu Met Gly Tyr Thr Glu Pro Asp Pro Arg Asp
            660                 665                 670

Asp Leu Ser Gly Met Asp Val Ala Arg Lys Leu Leu Ile Leu Ala Arg
        675                 680                 685

Glu Thr Gly Arg Glu Leu Glu Leu Ala Asp Ile Glu Ile Glu Pro Val
    690                 695                 700
```

Leu Pro Ala Glu Phe Asn Ala Glu Gly Asp Val Ala Ala Phe Met Ala
705 710 715 720

Asn Leu Ser Gln Leu Asp Asp Leu Phe Ala Ala Arg Val Ala Lys Ala
725 730 735

Arg Asp Glu Gly Lys Val Leu Arg Tyr Val Gly Asn Ile Asp Glu Asp
740 745 750

Gly Val Cys Arg Val Lys Ile Ala Glu Val Asp Gly Asn Asp Pro Leu
755 760 765

Phe Lys Val Lys Asn Gly Glu Asn Ala Leu Ala Phe Tyr Ser His Tyr
770 775 780

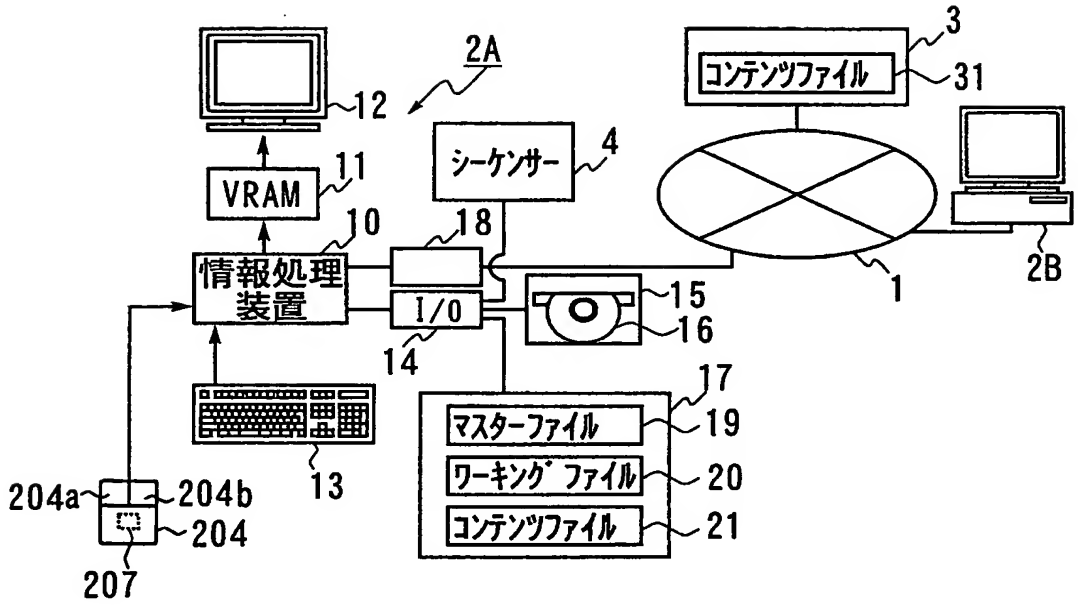
Tyr Gln Pro Leu Pro Leu Val Leu Arg Gly Tyr Gly Ala Gly Asn Asp
785 790 795 800

Val Thr Ala Ala Gly Val Phe Ala Asp Leu Leu Arg Thr Leu Ser Trp
805 810 815

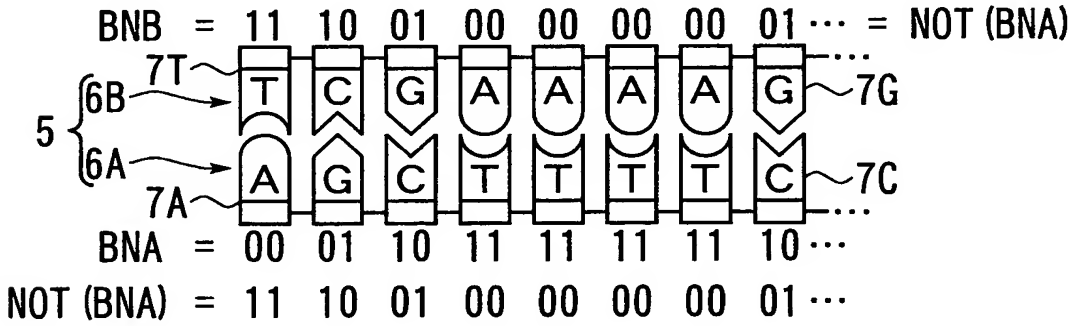
Lys Leu Gly Val
820

【書類名】　　図面

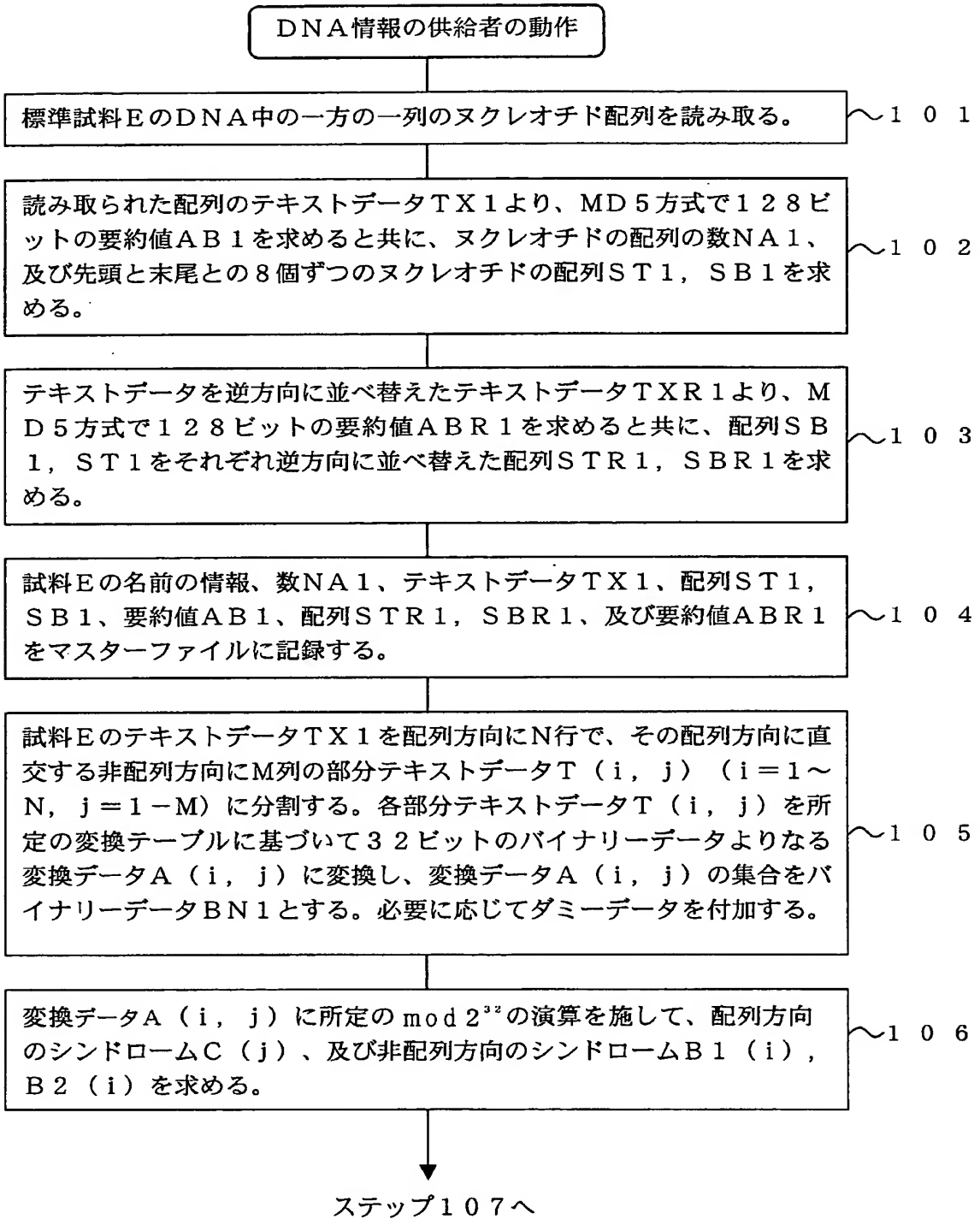
【図１】


2A
12
VRAM
11
10
情報処理装置
I/O
14
18
シーケンサー
4
3
コンテンツファイル
31
1
2B
15
16
17
マスターファイル
19
ワーキング ファイル
20
コンテンツファイル
21
13
204a
204b
204
207

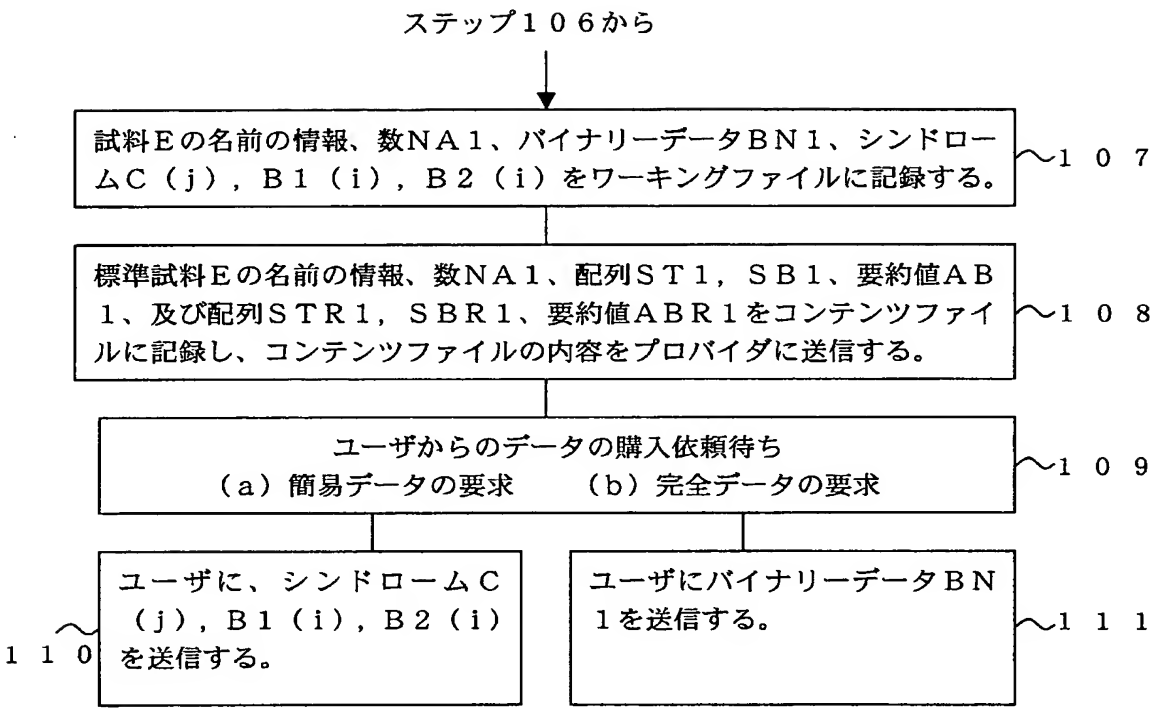
【図２】


BNB = 11 10 01 00 00 00 00 01 … = NOT (BNA)
7T
6B
5
6A
7A
T C G A A A A G
A G C T T T T C
7G
7C
BNA = 00 01 10 11 11 11 11 10 …
NOT (BNA) = 11 10 01 00 00 00 00 01 …

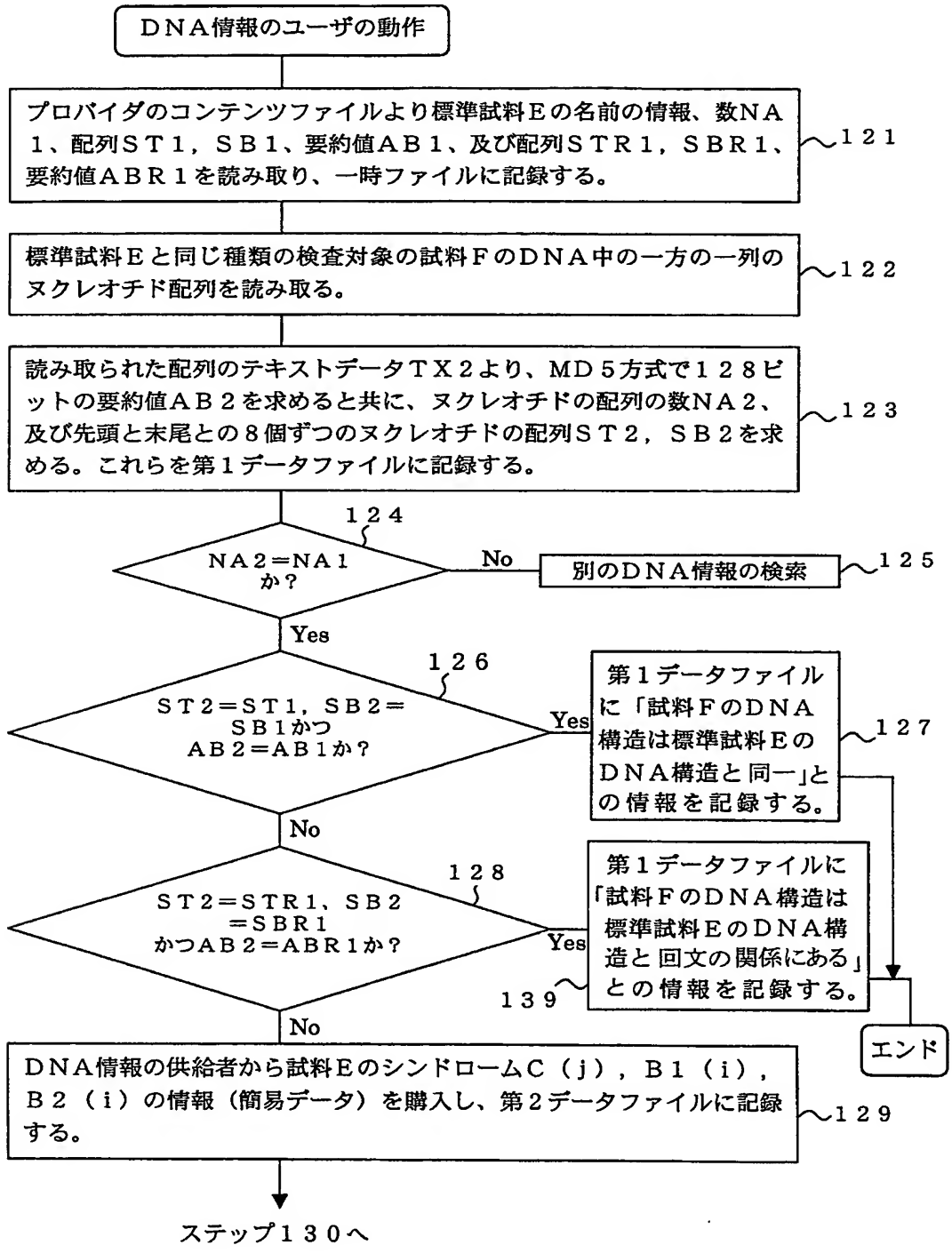
【図３】


ＤＮＡ情報の供給者の動作
標準試料ＥのＤＮＡ中の一方の一列のヌクレオチド配列を読み取る。
１０１
読み取られた配列のテキストデータＴＸ１より、ＭＤ５方式で１２８ビットの要約値ＡＢ１を求めると共に、ヌクレオチドの配列の数ＮＡ１、及び先頭と末尾との８個ずつのヌクレオチドの配列ＳＴ１，ＳＢ１を求める。
１０２
テキストデータを逆方向に並べ替えたテキストデータＴＸＲ１より、ＭＤ５方式で１２８ビットの要約値ＡＢＲ１を求めると共に、配列ＳＢ１，ＳＴ１をそれぞれ逆方向に並べ替えた配列ＳＴＲ１，ＳＢＲ１を求める。
１０３
試料Ｅの名前の情報、数ＮＡ１、テキストデータＴＸ１、配列ＳＴ１，ＳＢ１、要約値ＡＢ１、配列ＳＴＲ１，ＳＢＲ１、及び要約値ＡＢＲ１をマスターファイルに記録する。
１０４
試料ＥのテキストデータＴＸ１を配列方向にＮ行で、その配列方向に直交する非配列方向にＭ列の部分テキストデータＴ（ｉ，ｊ）（ｉ＝１～Ｎ，ｊ＝１－Ｍ）に分割する。各部分テキストデータＴ（ｉ，ｊ）を所定の変換テーブルに基づいて３２ビットのバイナリーデータよりなる変換データＡ（ｉ，ｊ）に変換し、変換データＡ（ｉ，ｊ）の集合をバイナリーデータＢＮ１とする。必要に応じてダミーデータを付加する。
１０５
変換データＡ（ｉ，ｊ）に所定のｍｏｄ2^{32}の演算を施して、配列方向のシンドロームＣ（ｊ）、及び非配列方向のシンドロームＢ１（ｉ），Ｂ２（ｉ）を求める。
１０６
ステップ１０７へ

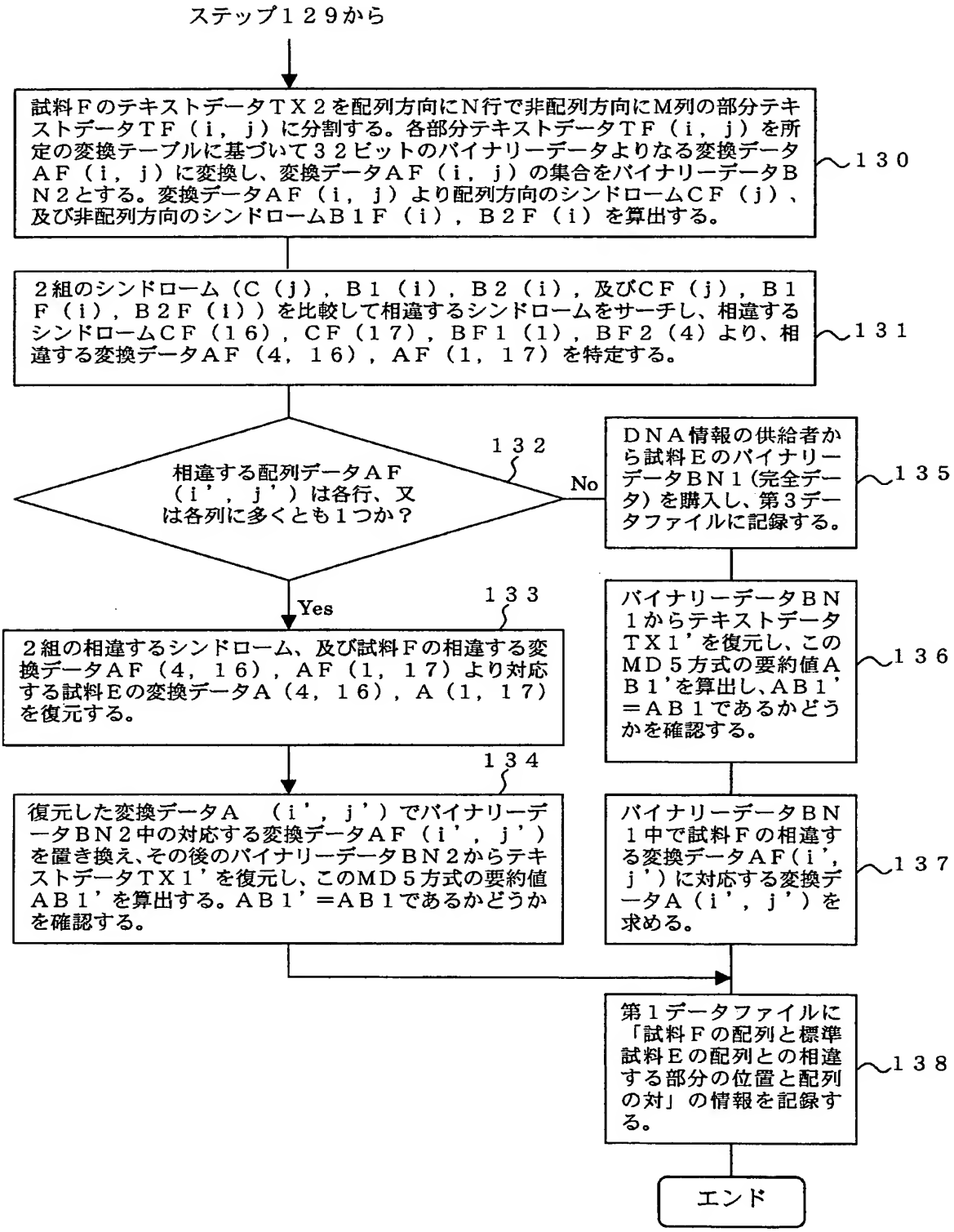
【図４】


ステップ１０６から
試料Ｅの名前の情報、数ＮＡ１、バイナリーデータＢＮ１、シンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）をワーキングファイルに記録する。
１０７
標準試料Ｅの名前の情報、数ＮＡ１、配列ＳＴ１，ＳＢ１、要約値ＡＢ１、及び配列ＳＴＲ１，ＳＢＲ１、要約値ＡＢＲ１をコンテンツファイルに記録し、コンテンツファイルの内容をプロバイダに送信する。
１０８
ユーザからのデータの購入依頼待ち
（ａ）簡易データの要求　（ｂ）完全データの要求
１０９
ユーザに、シンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）を送信する。
１１０
ユーザにバイナリーデータＢＮ１を送信する。
１１１

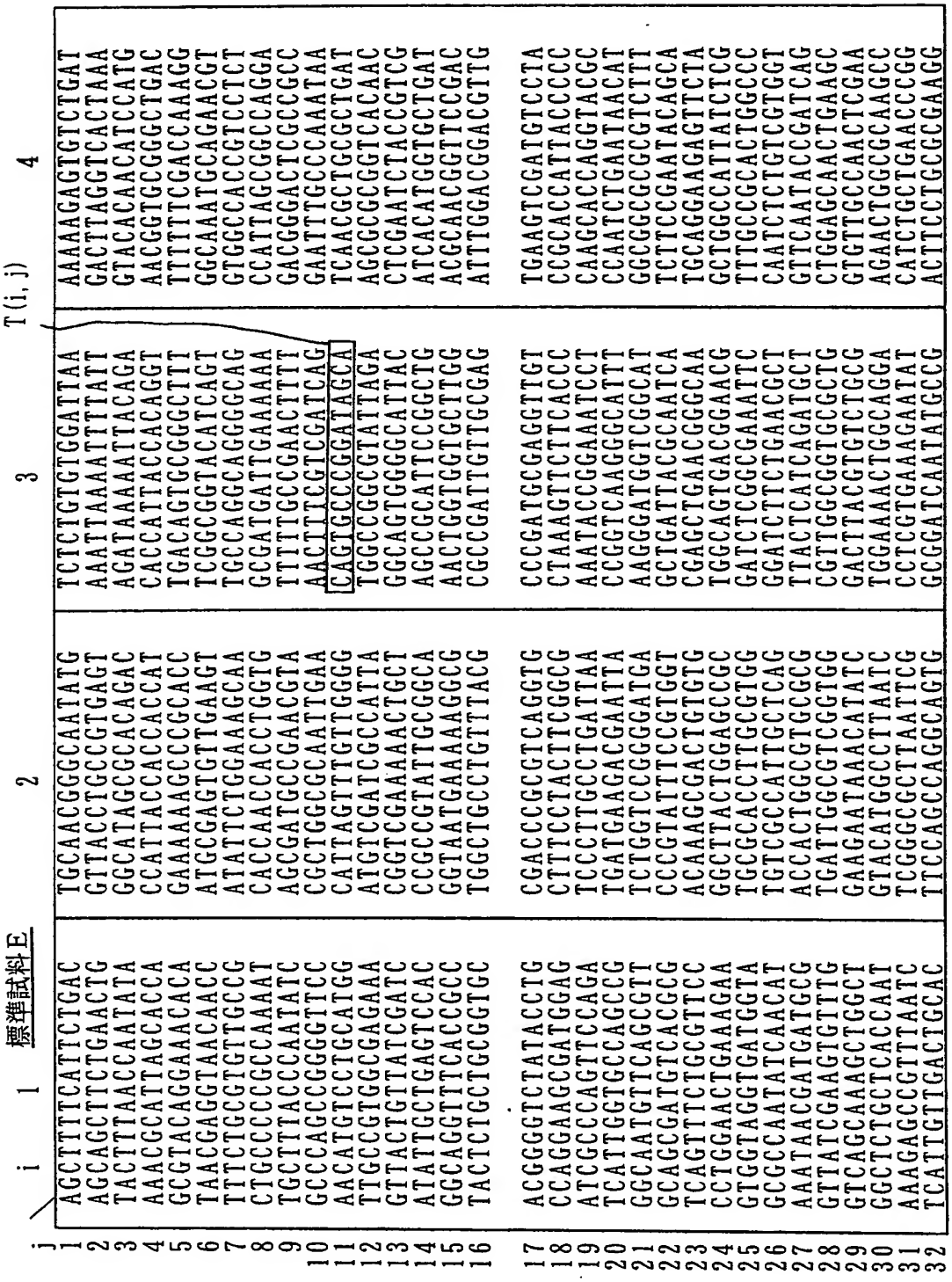
【図５】


ＤＮＡ情報のユーザの動作
プロバイダのコンテンツファイルより標準試料Ｅの名前の情報、数ＮＡ１、配列ＳＴ１，ＳＢ１、要約値ＡＢ１、及び配列ＳＴＲ１，ＳＢＲ１、要約値ＡＢＲ１を読み取り、一時ファイルに記録する。
１２１
標準試料Ｅと同じ種類の検査対象の試料ＦのＤＮＡ中の一方の一列のヌクレオチド配列を読み取る。
１２２
読み取られた配列のテキストデータＴＸ２より、ＭＤ５方式で１２８ビットの要約値ＡＢ２を求めると共に、ヌクレオチドの配列の数ＮＡ２、及び先頭と末尾との８個ずつのヌクレオチドの配列ＳＴ２，ＳＢ２を求める。これらを第１データファイルに記録する。
１２３
１２４
ＮＡ２＝ＮＡ１か？
No
別のＤＮＡ情報の検索
１２５
Yes
１２６
ＳＴ２＝ＳＴ１，ＳＢ２＝ＳＢ１かつＡＢ２＝ＡＢ１か？
Yes
第１データファイルに「試料ＦのＤＮＡ構造は標準試料ＥのＤＮＡ構造と同一」との情報を記録する。
１２７
No
１２８
ＳＴ２＝ＳＴＲ１，ＳＢ２＝ＳＢＲ１かつＡＢ２＝ＡＢＲ１か？
Yes
第１データファイルに「試料ＦのＤＮＡ構造は標準試料ＥのＤＮＡ構造と回文の関係にある」との情報を記録する。
１３９
No
エンド
ＤＮＡ情報の供給者から試料ＥのシンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）の情報（簡易データ）を購入し、第２データファイルに記録する。
１２９
ステップ１３０へ

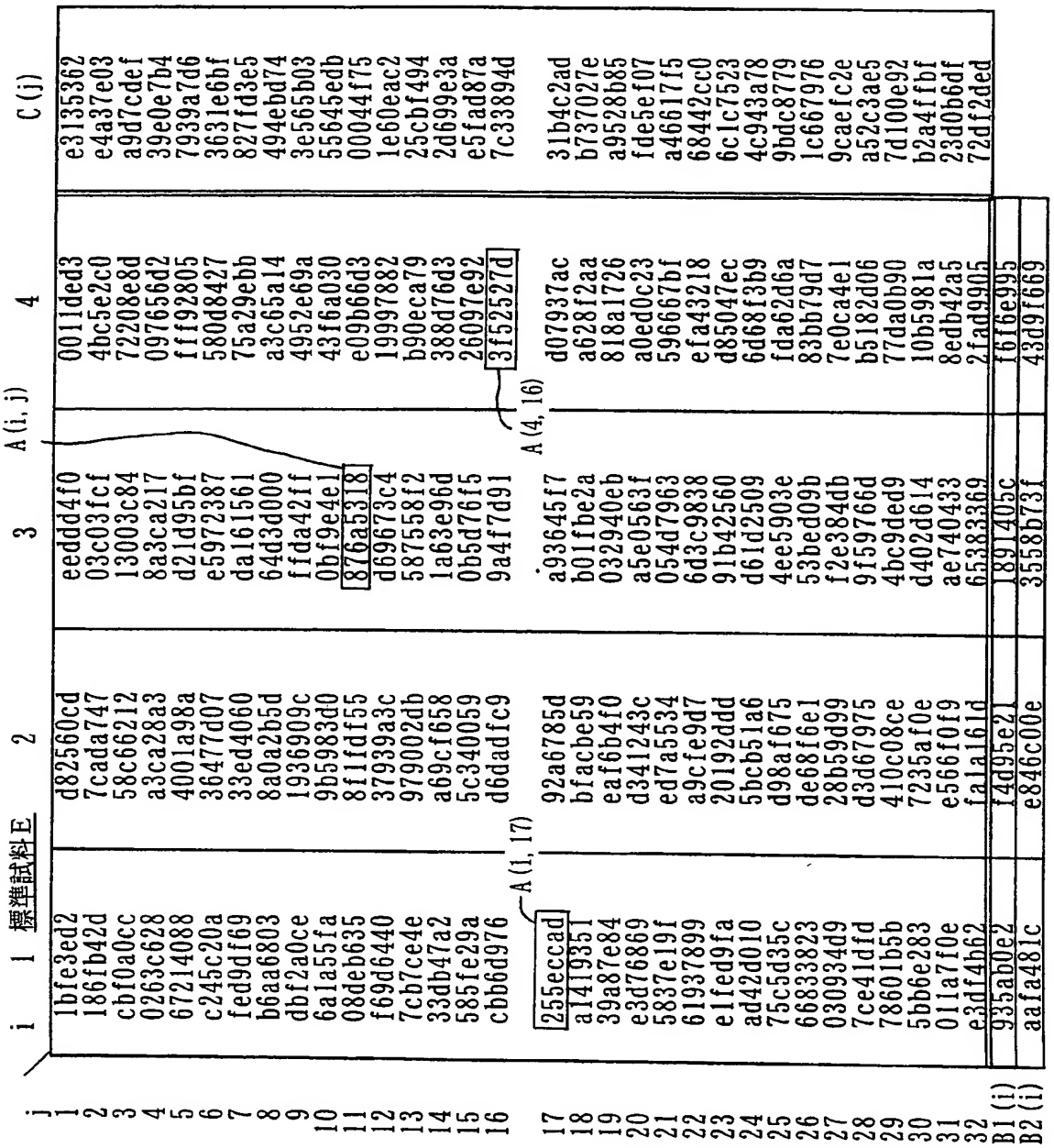
【図６】


ステップ１２９から
試料Ｆのテキストデータ ＴＸ２を配列方向にＮ行で非配列方向にＭ列の部分テキストデータＴＦ（ｉ，ｊ）に分割する。各部分テキストデータＴＦ（ｉ，ｊ）を所定の変換テーブルに基づいて３２ビットのバイナリーデータよりなる変換データＡＦ（ｉ，ｊ）に変換し、変換データＡＦ（ｉ，ｊ）の集合をバイナリーデータＢＮ２とする。変換データＡＦ（ｉ，ｊ）より配列方向のシンドロームＣＦ（ｊ）、及び非配列方向のシンドロームＢ１Ｆ（ｉ），Ｂ２Ｆ（ｉ）を算出する。
１３０
２組のシンドローム（Ｃ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ），及びＣＦ（ｊ），Ｂ１Ｆ（ｉ），Ｂ２Ｆ（ｉ））を比較して相違するシンドロームをサーチし、相違するシンドロームＣＦ（１６），ＣＦ（１７），ＢＦ１（１），ＢＦ２（４）より、相違する変換データＡＦ（４，１６），ＡＦ（１，１７）を特定する。
１３１
１３２
相違する配列データＡＦ（ｉ’，ｊ’）は各行、又は各列に多くとも１つか？
No
Yes
１３５
ＤＮＡ情報の供給者から試料Ｅのバイナリーデータ ＢＮ１（完全データ）を購入し、第３データファイルに記録する。
１３３
２組の相違するシンドローム、及び試料Ｆの相違する変換データＡＦ（４，１６），ＡＦ（１，１７）より対応する試料Ｅの変換データＡ（４，１６），Ａ（１，１７）を復元する。
１３６
バイナリーデータＢＮ１からテキストデータＴＸ１’を復元し、このＭＤ５方式の要約値ＡＢ１’を算出し、ＡＢ１’＝ＡＢ１であるかどうかを確認する。
１３４
復元した変換データＡ （ｉ’，ｊ’）でバイナリーデータＢＮ２中の対応する変換データＡＦ（ｉ’，ｊ’）を置き換え、その後のバイナリーデータＢＮ２からテキストデータＴＸ１’を復元し、このＭＤ５方式の要約値ＡＢ１’を算出する。ＡＢ１’＝ＡＢ１であるかどうかを確認する。
１３７
バイナリーデータＢＮ１中で試料Ｆの相違する変換データＡＦ（ｉ’，ｊ’）に対応する変換データＡ（ｉ’，ｊ’）を求める。
１３８
第１データファイルに「試料Ｆの配列と標準試料Ｅの配列との相違する部分の位置と配列の対」の情報を記録する。
エンド

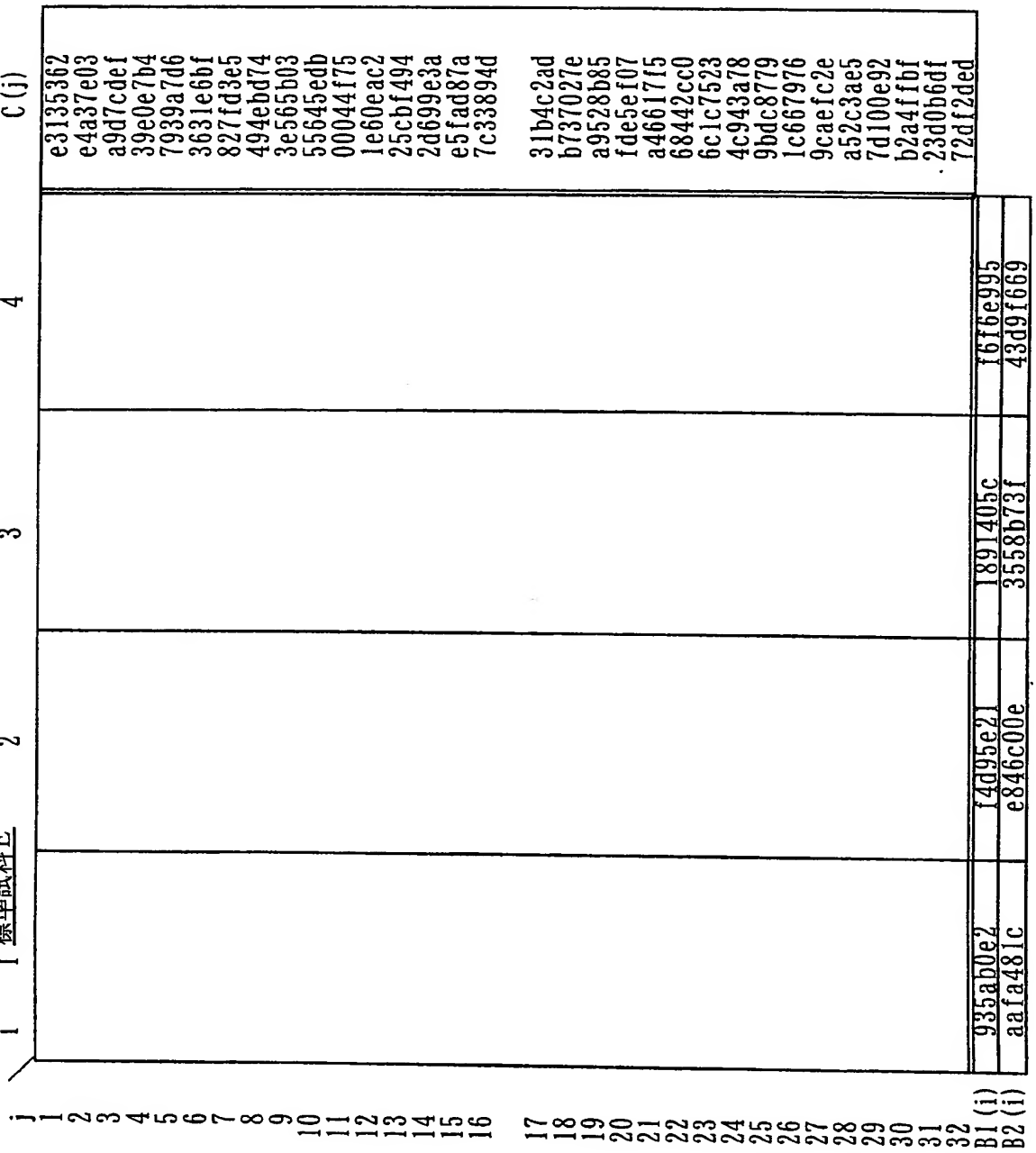
【図７】



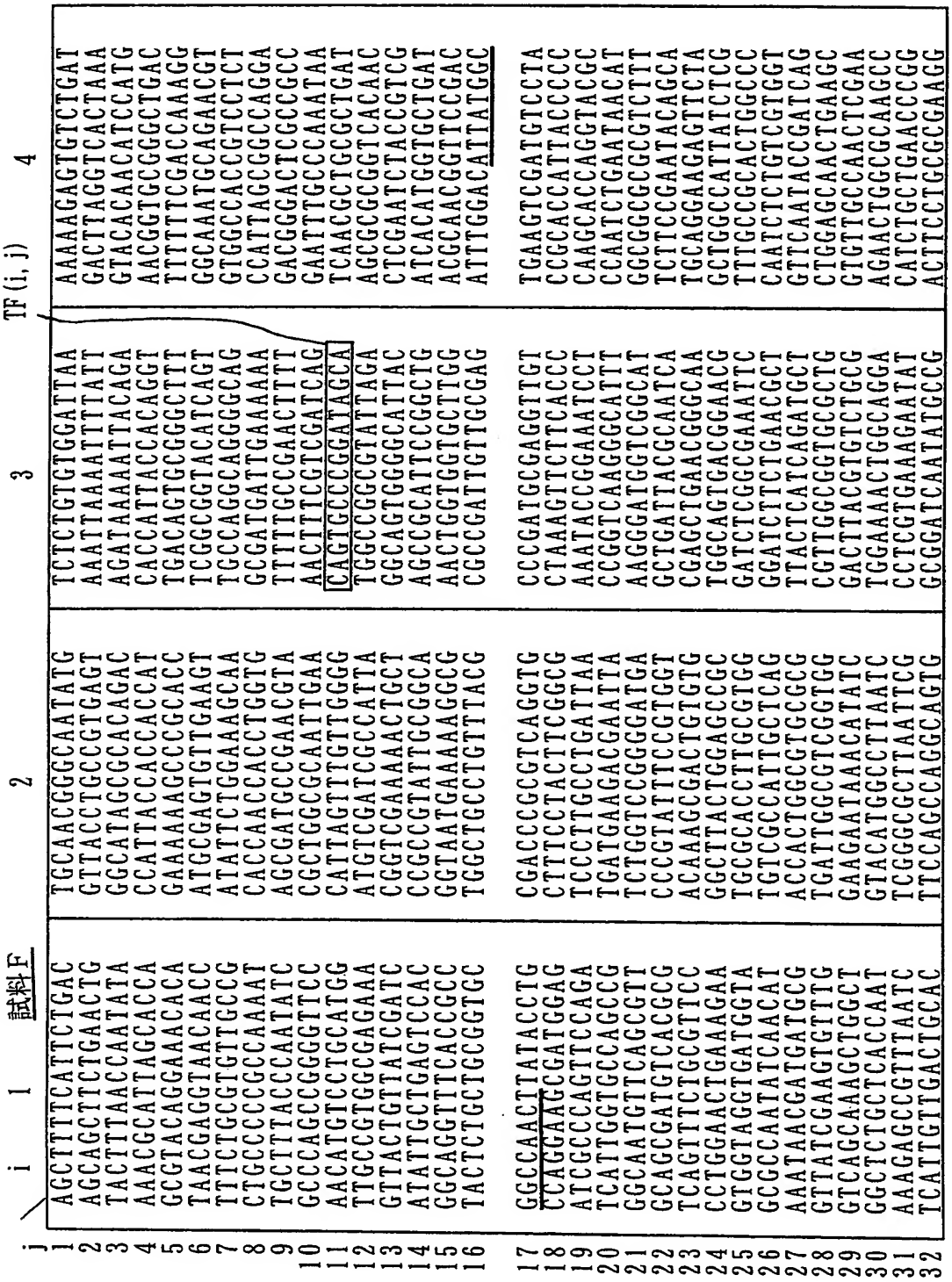
【図８】


i 1 標準試料E 2 3 A(i, j) 4 C(j)
j
1 1bfe3ed2 d82560cd eeddd4f0 0011ded3 e3135362
2 186fb42d 7cada747 03c03fcf 4bc5e2c0 e4a37e03
3 cbf0a0cc 58c66212 13003c84 72208e8d a9d7cdef
4 0263c628 a3ca28a3 8a3ca217 097656d2 39e0e7b4
5 67214088 4001a98a d21d95bf f1f92805 7939a7d6
6 c245c20a 36477d07 e5972387 580d8427 3631e6bf
7 fed9df69 33ed4060 da161561 75a29ebb 827fd3e5
8 b6aa6803 8a0a2b5d 64d3d000 a3c65a14 494ebd74
9 dbf2a0ce 1936909c ffda42ff 4952e69a 3e565b03
10 6a1a55fa 9b5983d0 0bf9e4e1 43f6a030 55645edb
11 08deb635 8f1fdf55 876a5318 e09b66d3 00044f75
12 f69d6440 37939a3c d69673c4 19997882 1e60eac2
13 7cb7ce4e 979002db 58755 8f2 b90eca79 25cbf494
14 33db47a2 a69cf658 1a63e96d 388d76d3 2d699e3a
15 585fe29a 5c340059 0b5d76f5 26097e92 e5fad87a
16 cbb6d976 d6dadfc9 9a4f7d91 3f52527d 7c33894d
A(1, 17) A(4, 16)
17 255eccad 92a6785d a93645f7 d07937ac 31b4c2ad
18 a1419351 bfacbe59 b01fbe2a a628f2aa b737027e
19 39a87e84 eaf6b4f0 032940eb 818a1726 a9528b85
20 e3d76869 d341243c a5e0563f a0ed0c23 fde5ef07
21 5837e19f ed7a5534 054d7963 596667bf a46617f5
22 61937899 a9cfe9d7 6d3c9838 efa43218 68442cc0
23 e1fed9fa 20192ddd 91b42560 d85047ec 6c1c7523
24 ad42d010 5bcb51a6 d61d2509 6d68f3b9 4c943a78
25 75c5d35c d98af675 4ee5903e fda62d6a 9bdc8779
26 66833823 de68f6e1 53bed09b 83bb79d7 1c667976
27 030934d9 28b59d99 f2e384db 7e0ca4e1 9caefc2e
28 7ce41dfd d3d67975 9f59766d b5182d06 a52c3ae5
29 78601b5b 410c08ce 4bc9ded9 77da0b90 7d100e92
30 5bb6e283 7235af0e d402d614 10b5981a b2a4ffbf
31 011a7f0e e566f0f9 ae740433 8edb42a5 23d0b6df
32 e3df4b62 fa1a161d 65383369 2fad9905 72df2ded
B1 (i) 935ab0e2 f4d95e21 189140 5c f6f6e995
B2 (i) aafa481c e846c00e 3558b73f 43d9f669

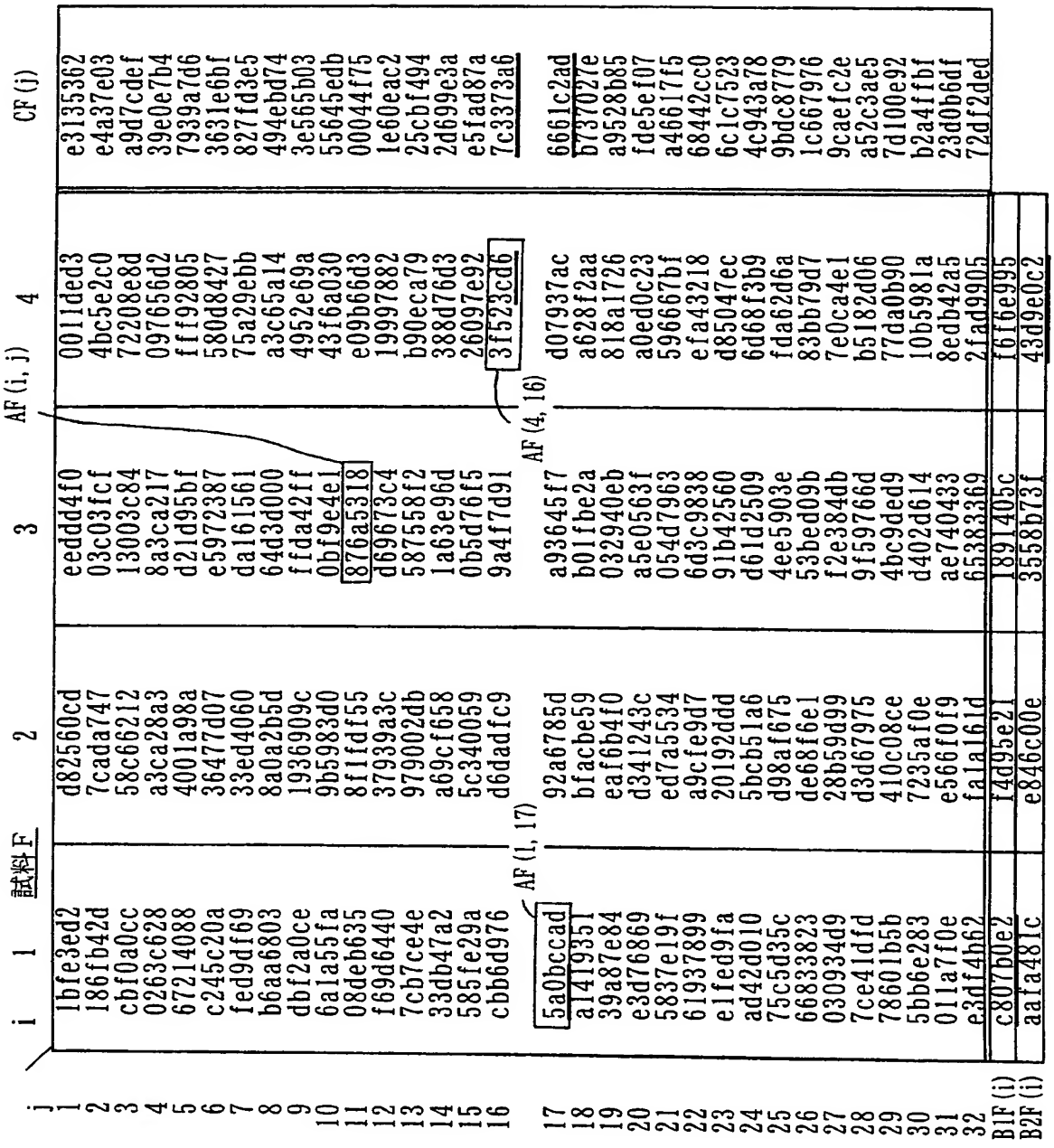
【図９】


i
1 標準試料E
2
3
4
C(j)
j
B1(i)
B2(i)
935ab0e2
aafa481c
f4d95e21
e846c00e
1891405c
3558b73f
f6f6e995
43d9f669
e3135362
e4a37e03
a9d7cdef
39e0e7b4
7939a7d6
3631e6bf
827fd3e5
494ebd74
3e565b03
55645edb
00044f75
1e60eac2
25cbf494
2d699e3a
e5fad87a
7c33894d
31b4c2ad
b737027e
a9528b85
fde5ef07
a4661715
68442cc0
6c1c7523
4c943a78
9bdc8779
1c667976
9caefc2e
a52c3ae5
7d100e92
b2a4f1bf
23d0b6df
72df2ded

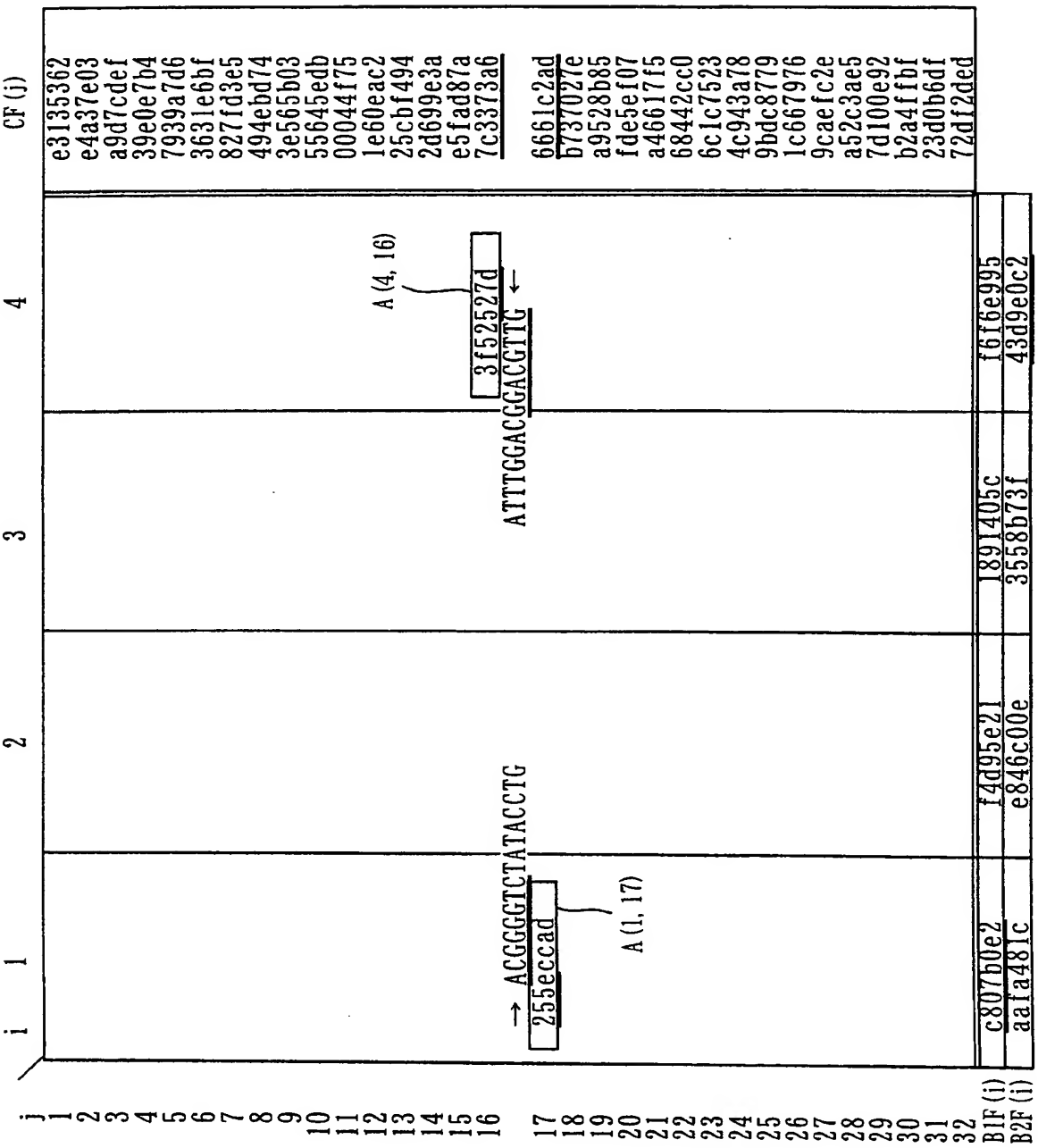
【図１０】


i
1
試料F
2
3
TF(i, j)
4
j
1 AGCTTTTCATTCTGAC TGCAACGGGCAATATG TCTCTGTGTGGATTAA AAAAAGAGTGTCTGAT
2 AGCAGCTTCTGAACTG GTTACCTGCCGTGAGT AAATTAAAATTTATT GACTTAGGTCACTAAA
3 TACTTTAACCAATATA GGCATAGCGCACAGAC AGATAAAAATTACAGA GTACACAACATCCATG
4 AAACGCATTAGCACCA CCATTACCACCACCAT CACCATTACCACAGGT AACGGTGCCGGGCTGAC
5 GCGTACAGGAAACACA GAAAAAAGCCCCGCACC TGACAGTGCCGGGCTTT TTTTTCGACCAAAGG
6 TAACGAGGTAACAACC ATGCGAGTGTTGAAGT TCGGCGGTACATCAGT GGCAAATGCAGAACGT
7 TTTCTGCGTGTTGCCG ATATTCTGGAAAGCAA TGCCAGGCAGGGCAG GTGGCCACCGTCCTCT
8 CTGCCCCCGCCAAAAT CACCAACCACCTGGTG GCGATGATTGAAAAAA CCATTAGCGGCCAGGA
9 TGCTTTACCCAATATC AGCGATGCCGAACGTA TTTTTGCCGAACTTTT GACGGGACTCGCCGCC
10 GCCCAGCCGGGTTCC CGCTGGCGCAATTGAA AACTTTCGTCGATCAG GAATTTGCCCCAAATAA
11 AACATGTCCTGCATGG CATTAGTTTGTTGGGG CAGTGCCCGGATAGCA TCAACGCTGCGCTGAT
12 TTGCCGTGGCGAGAAA ATGTCGATCGCCATTA TGGCCGGCGTATTAGA AGCGGCGGTCACAAC
13 GTTACTGTTATCGATC CGGTCGAAAAACTGCT GGCAGTGGGGCATTAC CTCGAATCTACCGTCG
14 ATATTGCTGAGTCCAC CCGCCGTATTGCGGCA AGCCGCATTCCGGCTG ATCACATGGTGCTGAT
15 GGCAGGTTTCACCGCC GGTAATGAAAAAGGCG AACTGGTGGTGCTTGG ACGCAACGGTTCCGAC
16 TACTCTGCTGCGGTGC TGGCTGCCTGTTTACG CGCCGATTGTTGTGCGAG ATTTGGACATTATGGC
17 GGCCAACTTATACCTG CGACCCGCGTCAGGTG CCCGATGCGAGGTTGT TGAAGTCGATGTCCTA
18 TCAGGAAGCGATGGAG CTTTCCTACTTCGGCG CTAAAGTTCTTCACCC CCGCACCATTACCCCC
19 ATCGCCCAGTTCCAGA TCCCTTGCCTGATTAA AAATACCGGAAATCCT CAAGCACCAGGTACGC
20 TCATTGGTGCCAGCCG TGATGAAGACGAATTA CCGGTCAAGGGCATTT CCAATCTGAATAACAT
21 GGCAATGTTCAGCGTT TCTGGTCCGGGGATGA AAGGGATGGTCGGCAT GGCGGCGCGCGTCTTT
22 GCAGCGATGTCACGCG CCCGTATTTCCGTGGT GCTGATTACGCAATCA TCTTCCGAATACAGCA
23 TCAGTTCTGCGTTCC ACAAAGCGACTGTGTG CGAGCTGAACGGGCAA TGCAGGAAGAGTTCTA
24 CCTGGAACTGAAAGAA GGCTTACTGGAGCCGC TGGCAGTGACGGAACG GCTGGCCATTATCTCG
25 GTGGTAGGTGATGGTA TGCGCACCTTGCGTGG GATCTCGGCGAAATTC TTTGCCGCACTGGCCC
26 GCGCCAATATCAACAT TGTCGCCATTGCTCAG GGATCTTCTGAACGCT CAATCTCTGTCGTGGT
27 AAATAACGATGATGCG ACCACTGGCGTGCGCG TTACTCATCAGATGCT GTTCAATACCGATCAG
28 GTTATCGAAGTGTTTG TGATTGGCGTCGGTGG CGTTGGCGGTGCGCTG CTGGAGCAACTGAAGC
29 GTCAGCAAAGCTGGCT GAAGAATAAACATATC GACTTACGTGTCTGCG GTGTTGCCAACTCGAA
30 GGCTCTGCCTCACCAAT GTACATGGCCTTAATC TGGAAAACTGGCAGGA AGAACTGGCGCAAGCC
31 AAAGAGCCGTTTAATC TCGGGCGCTTAATTCG CCTCGTGAAAGAATAT CATCTGCTGAACCCGG
32 TCATTGTTGACTGCAC TTCCAGCCAGGCAGTG GCGGATCAATATGCCG ACTTCCTGCGCGAAGG

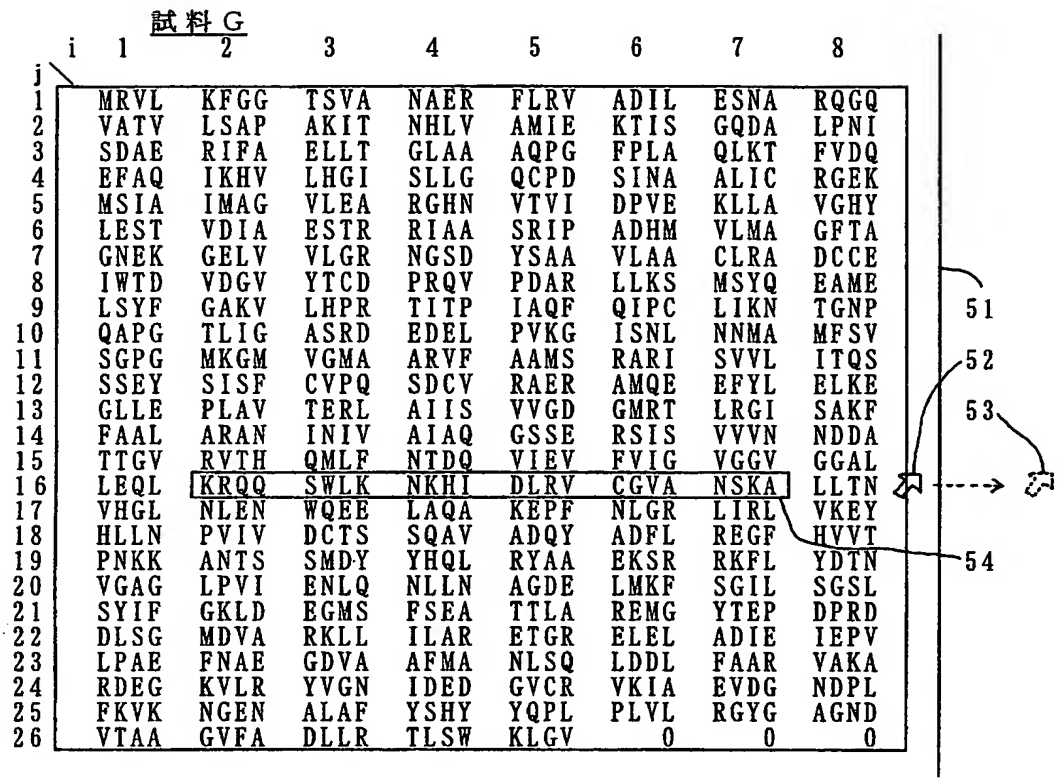
【図１１】


AF(i, j)
AF(1, 17)
AF(4, 16)
試料F
i
j
1 2 3 4 CF(j)
1bfe3ed2 d82560cd eeddd4f0 0011ded3 e3135362
186fb42d 7cada747 03c03fcf 4bc5e2c0 e4a37e03
cbf0a0cc 58c66212 13003c84 72208e8d a9d7cdef
0263c628 a3ca28a3 8a3ca217 097656d2 39e0e7b4
67214088 4001a98a d21d95bf fff92805 7939a7d6
c245c20a 36477d07 e5972387 580d8427 3631e6bf
fed9df69 33ed4060 da161561 75a29ebb 827fd3e5
b6aa6803 8a0a2b5d 64d3d000 a3c65a14 494ebd74
dbf2a0ce 1936909c ffda42ff 4952e69a 3e565b03
6a1a55fa 9b5983d0 0bf9e4e1 43f6a030 55645edb
08deb635 8f1fdf55 876a5318 e09b66d3 00044f75
f69d6440 37939a3c d69673c4 19997882 1e60eac2
7cb7ce4e 979002db 587558f2 b90eca79 25cbf494
33db47a2 a69cf658 1a63e96d 388d76d3 2d699e3a
585fe29a 5c340059 0b5d76f5 26097e92 e5fad87a
cbb6d976 d6dadfc9 9a4f7d91 3f523cd6 7c3373a6
5a0bccad 92a6785d a93645f7 d07937ac 6661c2ad
a1419351 bfacbe59 b01fbe2a a628f2aa b737027e
39a87e84 eaf6b4f0 032940eb 818a1726 a9528b85
e3d76869 d3412243c a5e0563f a0ed0c23 fde5ef07
5837e19f ed7a5534 054d7963 596667bf a46617f5
61937899 a9cfe9d7 6d3c9838 efa43218 68442cc0
e1fed9fa 20192ddd 91b42560 d85047ec 6c1c7523
ad42d010 5bcb51a6 d61d2509 6d68f3b9 4c943a78
75c5d35c d98af675 4ee5903e fda62d6a 9bdc8779
66833823 de68f6e1 53bed09b 83bb79d7 1c667976
030934d9 28b59d99 f2e384db 7e0ca4e1 9caefc2e
7ce41dfd d3d67975 9f59766d b5182d06 a52c3ae5
78601b5b 410c08ce 4bc9ded9 77da0b90 7d100e92
5bb6e283 7235af0e d402d614 10b5981a b2a4ffbf
011a7f0e e566f0f9 ae740433 8edb42a5 23d0b6df
e3df4b62 fa1a161d 65383369 2fad9905 72df2ded
B1F(i) c807b0e2 f4d95e21 1891405c f6f6e995
B2F(i) aafa481c e846c00e 3558b73f 43d9e0c2

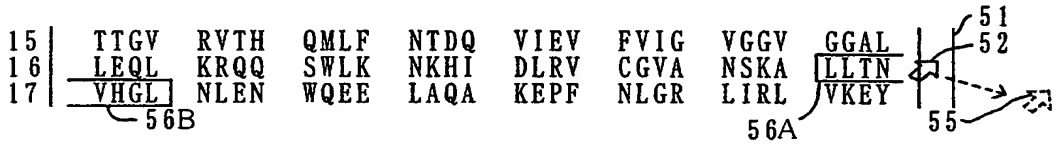
【図１２】


i
1
2
3
4
CF(j)
→ ACGGGGTCTATACCTG
255eccad
A(1, 17)
ATTTGGACGGACGTTG ←
3f52527d
A(4, 16)
e3135362
e4a37e03
a9d7cdef
39e0e7b4
7939a7d6
3631e6bf
827fd3e5
494ebd74
3e565b03
55645edb
00044f75
1e60eac2
25cbf494
2d699e3a
e5fad87a
7c3373a6
6661c2ad
b737027e
a9528b85
fde5ef07
a46617f5
68442cc0
6c1c7523
4c943a78
9bdc8779
1c667976
9caefc2e
a52c3ae5
7d100e92
b2a4ffbf
23d0b6df
72df2ded
c807b0e2
aafa481c
B1F(i)
f4d95e21
e846c00e
B2F(i)
1891405c
3558b73f
f6f6e995
43d9e0c2
j
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

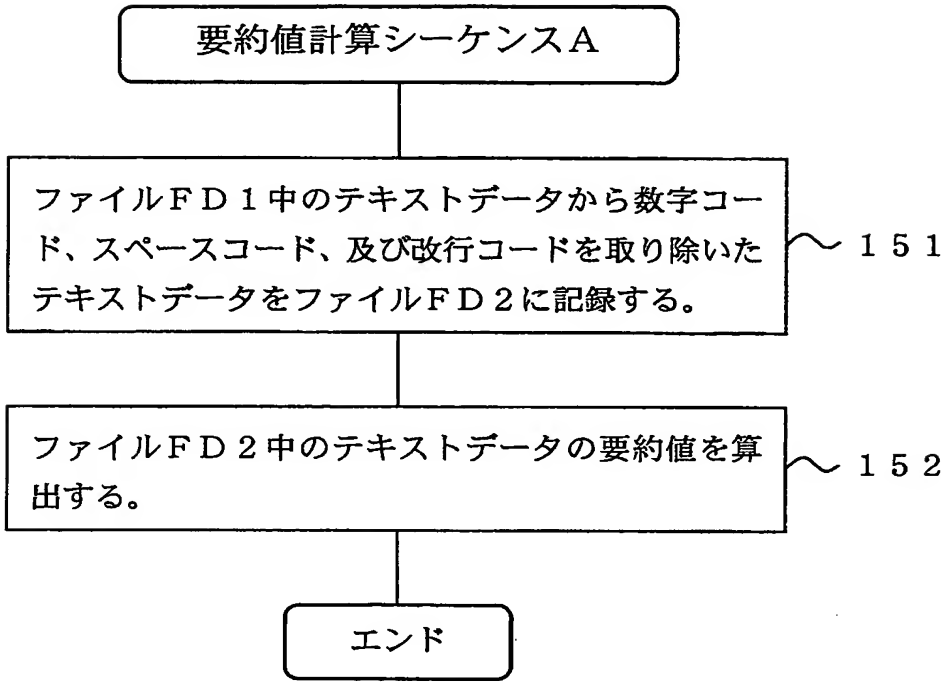
【図１３】


試料Ｇ
i 1 2 3 4 5 6 7 8
j
1 MRVL KFGG TSVA NAER FLRV ADIL ESNA RQGQ
2 VATV LSAP AKIT NHLV AMIE KTIS GQDA LPNI
3 SDAE RIFA ELLT GLAA AQPG FPLA QLKT FVDQ
4 EFAQ IKHV LHGI SLLG QCPD SINA ALIC RGEK
5 MSIA IMAG VLEA RGHN VTVI DPVE KLLA VGHY
6 LEST VDIA ESTR RIAA SRIP ADHM VLMA GFTA
7 GNEK GELV VLGR NGSD YSAA VLAA CLRA DCCE
8 IWTD VDGV YTCD PRQV PDAR LLKS MSYQ EAME
9 LSYF GAKV LHPR TITP IAQF QIPC LIKN TGNP
10 QAPG TLIG ASRD EDEL PVKG ISNL NNMA MFSV
11 SGPG MKGM VGMA ARVF AAMS RARI SVVL ITQS
12 SSEY SISF CVPQ SDCV RAER AMQE EFYL ELKE
13 GLLE PLAV TERL AIIS VVGD GMRT LRGI SAKF
14 FAAL ARAN INIV AIAQ GSSE RSIS VVVN NDDA
15 TTGV RVTH QMLF NTDQ VIEV FVIG VGGV GGAL
16 LEQL KRQQ SWLK NKHI DLRV CGVA NSKA LLTN
17 VHGL NLEN WQEE LAQA KEPF NLGR LIRL VKEY
18 HLLN PVIV DCTS SQAV ADQY ADFL REGF HVVT
19 PNKK ANTS SMDY YHQL RYAA EKSR RKFL YDTN
20 VGAG LPVI ENLQ NLLN AGDE LMKF SGIL SGSL
21 SYIF GKLD EGMS FSEA TTLA REMG YTEP DPRD
22 DLSG MDVA RKLL ILAR ETGR ELEL ADIE IEPV
23 LPAE FNAE GDVA AFMA NLSQ LDDL FAAR VAKA
24 RDEG KVLR YVGN IDED GVCR VKIA EVDG NDPL
25 FKVK NGEN ALAF YSHY YQPL PLVL RGYG AGND
26 VTAA GVFA DLLR TLSW KLGV 0 0 0
51
52
53
54

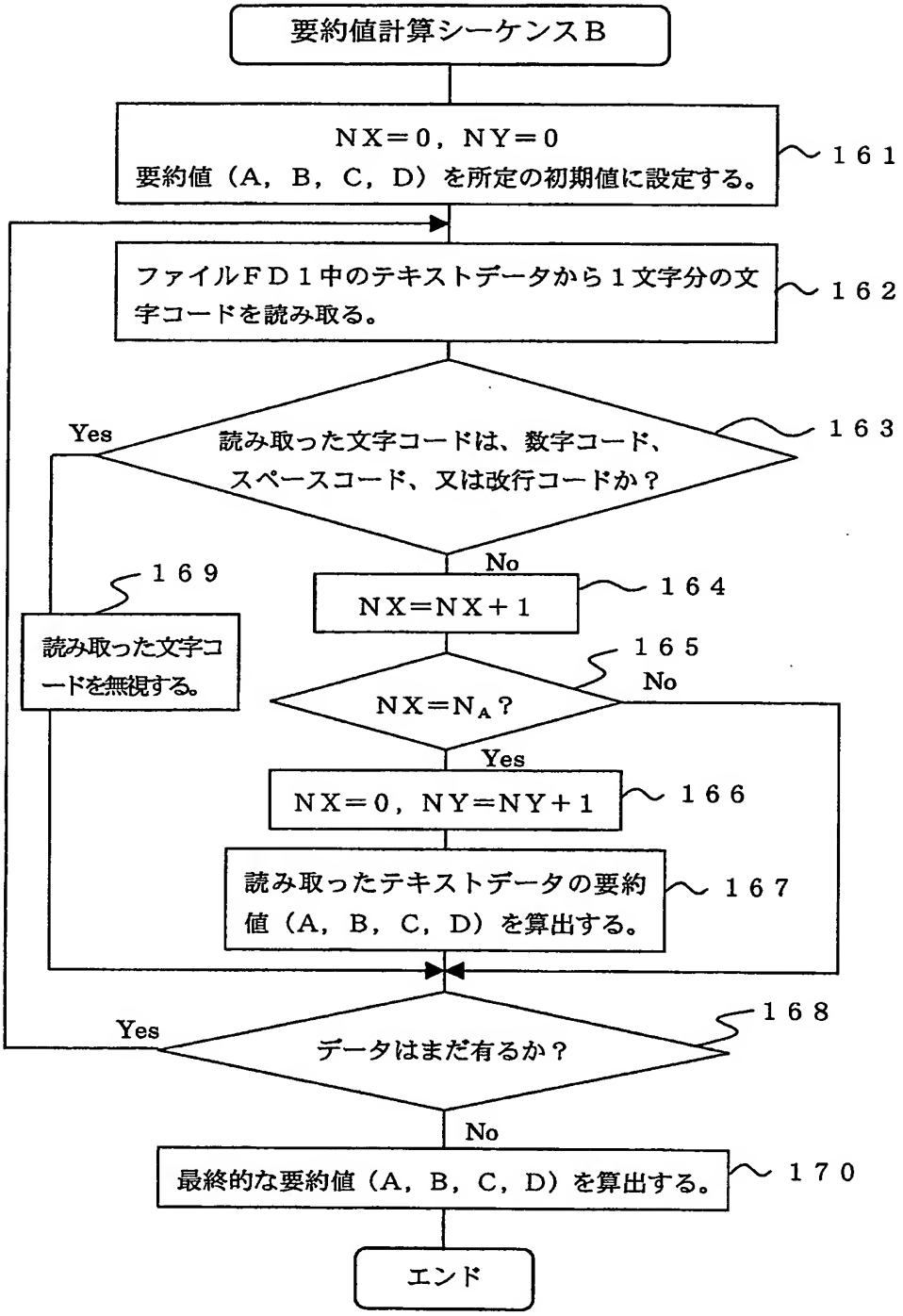
【図１４】


15 TTGV RVTH QMLF NTDQ VIEV FVIG VGGV GGAL
16 LEQL KRQQ SWLK NKHI DLRV CGVA NSKA LLTN
17 VHGL NLEN WQEE LAQA KEPF NLGR LIRL VKEY
56B
56A
51
52
55

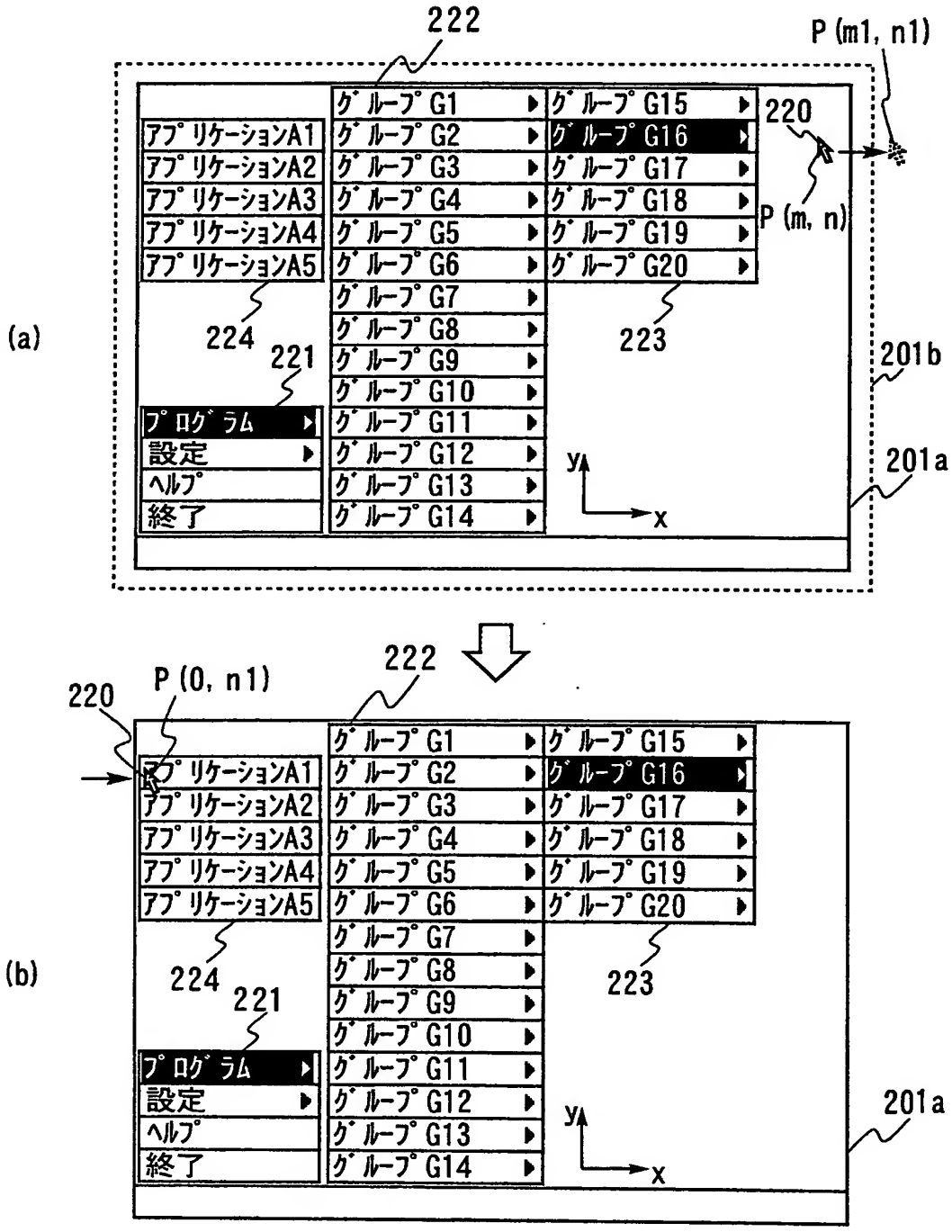
【図１５】


要約値計算シーケンスＡ
ファイルＦＤ１中のテキストデータから数字コード、スペースコード、及び改行コードを取り除いたテキストデータをファイルＦＤ２に記録する。
１５１
ファイルＦＤ２中のテキストデータの要約値を算出する。
１５２
エンド

【図１６】


要約値計算シーケンスＢ
ＮＸ＝０，ＮＹ＝０
要約値（Ａ，Ｂ，Ｃ，Ｄ）を所定の初期値に設定する。
１６１
ファイルＦＤ１中のテキストデータから１文字分の文字コードを読み取る。
１６２
読み取った文字コードは、数字コード、スペースコード、又は改行コードか？
１６３
Yes
No
１６９
読み取った文字コードを無視する。
ＮＸ＝ＮＸ＋１
１６４
１６５
ＮＸ＝Ｎ$_A$？
No
Yes
ＮＸ＝０，ＮＹ＝ＮＹ＋１
１６６
読み取ったテキストデータの要約値（Ａ，Ｂ，Ｃ，Ｄ）を算出する。
１６７
１６８
データはまだ有るか？
Yes
No
最終的な要約値（Ａ，Ｂ，Ｃ，Ｄ）を算出する。
１７０
エンド

【図１７】


(a)
222
P (m1, n1)
220
P (m, n)
224
221
223
201b
201a
y
x
アプリケーションA1
アプリケーションA2
アプリケーションA3
アプリケーションA4
アプリケーションA5
プログラム
設定
ヘルプ
終了
グループ G1
グループ G2
グループ G3
グループ G4
グループ G5
グループ G6
グループ G7
グループ G8
グループ G9
グループ G10
グループ G11
グループ G12
グループ G13
グループ G14
グループ G15
グループ G16
グループ G17
グループ G18
グループ G19
グループ G20
(b)
222
P (0, n1)
220
224
221
223
201a
y
x

【書類名】　　要約書

【要約】

【課題】　　核酸中のヌクレオチドの配列情報、又はタンパク質中のアミノ酸の配列情報をできるだけ少ないデータ量で記録する。

【解決手段】　試料のＤＮＡを構成する一方の一列のヌクレオチドの配列を示すテキストデータを所定の変換テーブルに従ってバイナリーデータに変換し、このバイナリーデータを複数行で複数列の変換データＡ（ｉ，ｊ）に分割する。各列の変換データＡ（ｉ，ｊ）を配列方向に演算してシンドロームＣ（ｊ）（ｊ＝１，２，…）を求め、各行の変換データＡ（ｉ，ｊ）を非配列方向に演算して２組のシンドロームＢ１（ｉ），Ｂ２（ｉ）（ｉ＝１，２，…）を求め、シンドロームＣ（ｊ），Ｂ１（ｉ），Ｂ２（ｉ）によってヌクレオチドの配列を近似的に表す。

【選択図】　　図８

出　願　人　履　歴　情　報

識別番号　　〔３００００５１３〕

１．変更年月日　　１９９９年１２月１７日
［変更理由］　新規登録
住　所　埼玉県浦和市西堀４丁目１１番７号６２７
氏　名　大森　聡

２．変更年月日　　２００１年　７月１３日
［変更理由］　住所変更
住　所　埼玉県さいたま市西堀４丁目１１番７号６２７
氏　名　大森　聡